

Design • Build • Program • Learn • Compete

SERVO

FOR THE ROBOT INNOVATOR
www.servomagazine.com

MAGAZINE
March 2013

**Shake
Hands
With The
MK2**

**Talk
To
Your
Robot
With
Light**

Don't Stress Out Your Bot
Object Identification
Can Help It Cope

Go New Mobile App!
Available For:
iOS - Android
Kindle Fire
SERVO Goes Where You



Grab the Spotlight with Hitec!

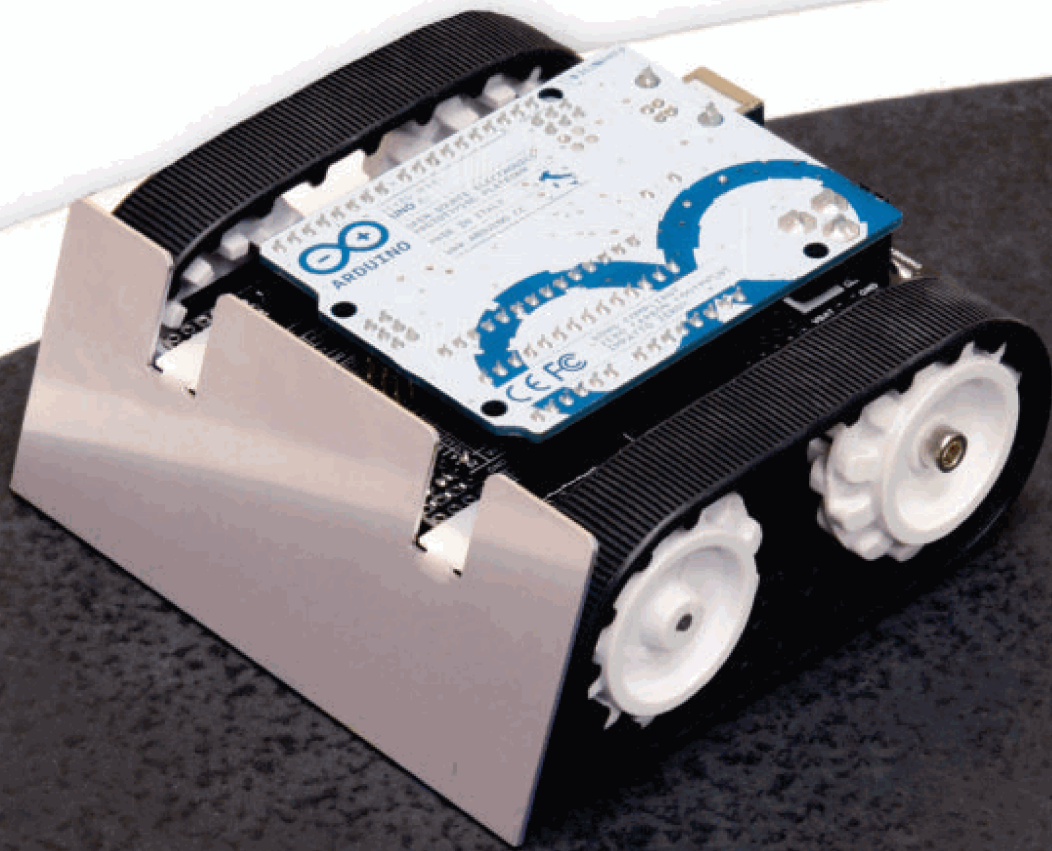


Hitec's superb line of radios and servos are the clear choice for robot enthusiasts at the top of their game. Our flagship **Aurora 9** computer radio with 2.4GHz Adaptive Frequency Hopping Spread Spectrum technology has unwavering reliability, endless programming capabilities and impressive, easy-to-master operation. The three-channel **Aggressor SRX Pro** pistol-grip radio with 10-model memory uses Direct Sequence Spread Spectrum technology to ensure its solid control signal. Choose one of these two or the **Eclipse 7 Pro**, **Optic 6 Sport**, **Optic 5** or **Aggressor 3DS**. No matter what your project, Hitec has the gear to keep you and your bot in the spotlight!



12115 Paine Street • Poway, CA 92064 • 858.748.6948 • www.hitecrd.com

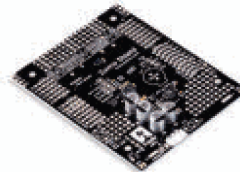
If you build it, it will push.



Basic Sumo Blade



Many Motor Options



Zumo Shield



Reflectance Sensors



Zumo Chassis Kit

Build and customize your Zumo.

Put your Arduino or compatible controller on the right tracks with the Zumo chassis and Arduino shield! The Zumo is a small, tracked robot platform that works with a variety of micro metal gearmotors to allow for a customizable combination of torque and speed. Add a Zumo shield, which includes a dual motor driver, buzzer, and three-axis accelerometer and compass, to make an Arduino-controlled robot that can really throw its weight around!

Columns

08 Robytes

by Jeff Eckert

Stimulating Robot Tidbits

- Beast of Burden Aces Field Tests
- High Speed Book Scanner
- Hovering Snoop Offers Endurance, Security
- Fiddler on the Beach

10 GeerHead

by David Geer

HDT Global's Adroit MK2 Robot Arms

They may not be able to transform into cars, trucks, or aircraft, but there is a lot of stuff these manipulators can do.

13 Ask Mr. Roboto

by Dennis Clark

Your Problems Solved Here

The saga on getting control over your FrankenRoomba continues ...

Departments

06 Mind/Iron

18 New Products

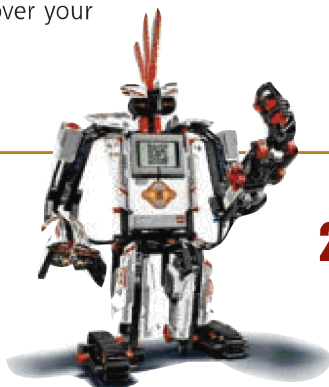
20 Events
Calendar

21 Showcase

64 SERVO
Webstore

80 Robo-Links

80 Advertiser's
Index



- Dextre is Pumped!
- An Amazing Opportunity
- Rise of the Phoenix
- Pet Project
- Face it, Kid
- Getting Under Our Skin
- Lynx — the "Un" Robot
- EV3 Unveiled

22 Bots in Brief



PAGE 67

67 Twin Tweaks

by Bryce and Evan Woolley

Robots in 4D

4D Systems is offering some new smart OLED and LCD display modules that will play well with your robots.

74 Then and Now

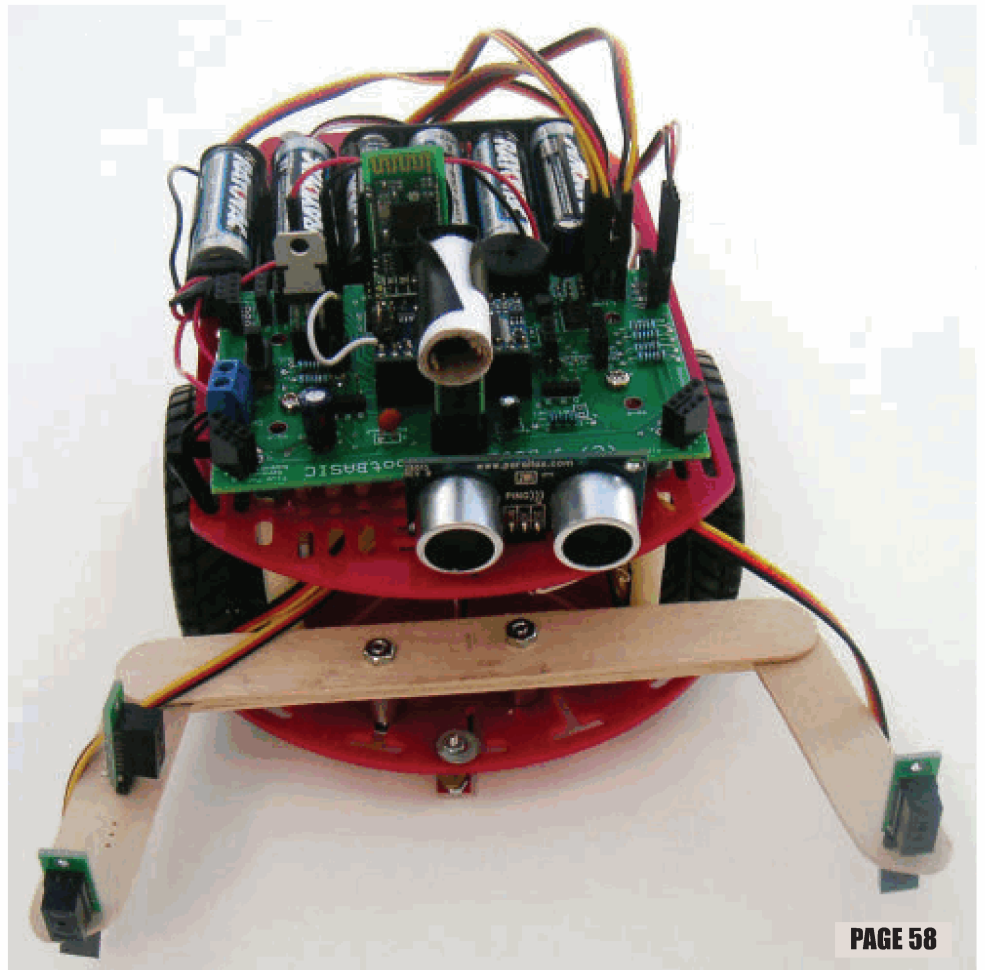
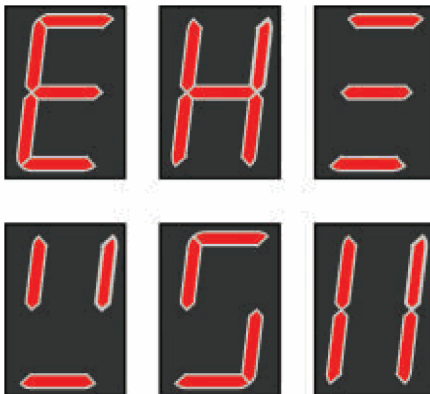
by Tom Carroll

Controlling Robots in the Air and on the Ground

Multi-rotor flyers are proving to be the next level of robotics.



PAGE 36



36 Talking With Light

by Gordon McComb

Using light to provide feedback for your robot can be as simple as a single LED, or more complex using multiple LED colors. In this article, we'll discuss simple techniques to communicate using light with your Arduino-based robot. The projects are simple to do, with only basic components and wiring needed.

44 Build the Kronos Flyer

by Michael Simpson

Part 5: Upgrades.

Read about some add-on items that hopefully will inspire your imagination.

54 The Neato Robot and ROS

by Boris Kudryavtsev

This eighth grader shows us how to hack a vacuum bot to turn it into a TABLEBot for competition.

58 Robotic Sensors

by John Blankenship and Samuel Mishal

By giving your robot just a rudimentary ability to identify objects, you can increase its ability to cope with more complex situations.

The Combat Zone...

26 BUILD REPORT:

Mini Bot Hockey Bots — Part 2

30 PARTS IS PARTS:

Product Review:

OrangeRx T-SIX 2.4 GHz Transmitter

32 Building a Large Scrolling Display — Part 2

Mind / Iron

by Bryan Bergeron, Editor

There's a Chip for That

Need a generic desk and chair by tomorrow morning? Head to IKEA. Want to learn cabinet making so that you can make a custom desk and chairs? Then get to work planning and stocking your workshop.

It's the same with circuits. If you need a compact, affordable circuit tomorrow that performs a common function, say, voltage doubling, then you'll likely find a dozen or more affordable ASICs (application specific integrated circuits) that will satisfy your requirements — with no or few external components.

However, if your goal is to learn how a particular circuit works and how it can be modified to suit your particular needs, then you should consider a circuit made of discrete components, transistors, and generic ICs. The end product will likely cost more in terms of component prices, printed circuit board real estate, and your time. However, you'll have an opportunity to learn more.

Of course, you have to make the most of this. If you simply insert the parts into a breadboard, solder everything into place, and then apply power, you'll waste the moment. For example, let's say you have a voltage doubler circuit based on the LM555 timer chip. You can change the frequency and duty cycle and external component values to assess how the changes affect output voltage and current.

The process vs. product tension isn't the only reason to consider single chip solutions over discrete component circuit design and construction. Some of the newest chips have exceptional specifications: higher efficiency, lower voltage requirements, less potential for interference, better thermal regulation, lower noise floor, etc. Sometimes it comes down to weight. For example, I've been working with quadcopter controllers, and every extra ounce of circuitry means about five minutes of less flying time, less maneuverability, and less space for something else.

Often, it's a matter of focus. Let's say you've built a LIDAR that requires a voltage doubler. If your focus is to learn how to apply LIDAR to robot navigation, then don't waste your time designing or building a low level supply circuit. Pick up a voltage doubler chip and get to work on the LIDAR proper.

I've found the major hurdle in using an ASIC over a handful of discrete components is determining if my wish-chip exists. My first go-to resource is Digi-Key. They maintain a good (albeit incomplete) catalog of special purpose chips at reasonable prices. My second stop is Mouser — another online supply company. Prices are usually a tad higher than Digi-Key, but they also handle a slightly different IC product line. Failing to find something in my top two list, I turn to the chip suppliers directly, such as TI and National Instruments. Most of these companies provide powerful search engines indexed to their products.

In closing, there's nothing inherently wrong with using discrete component circuits in your projects. It simply depends on what's best for your budget, the components you have on hand, your experience with new and old ICs, and whether the circuit is for your personal use or destined for the masses. However, make an informed choice. Check out which ICs or components best fit your needs. If, after a first exploration, you can't find the chip you're looking for, keep looking. New ASICs are announced daily.

SV

FOR THE
ROBOT
INNOVATOR

SERVO
MAGAZINE

Published Monthly By

T & L Publications, Inc.

430 Princeland Ct., Corona, CA 92879-1300

(951) 371-8497

FAX **(951) 371-3052**

Webstore Only **1-800-783-4624**

www.servomagazine.com

Subscriptions

Toll Free **1-877-525-2539**

Outside US **1-818-487-4545**

P.O. Box 15277, N. Hollywood, CA 91615

PUBLISHER

Larry Lemieux

publisher@servomagazine.com

**ASSOCIATE PUBLISHER/
VP OF SALES/MARKETING**

Robin Lemieux

display@servomagazine.com

EDITOR

Bryan Bergeron

techedit-servo@yahoo.com

CONTRIBUTING EDITORS

Jeff Eckert

Jenn Eckert

Tom Carroll

Kevin Berry

Dennis Clark

R. Steven Rainwater

Michael Simpson

David Geer

Bryce Woolley

Evan Woolley

Gordon McComb

Boris Kudryavtsev

John Blankenship

Samuel Mishal

Pete Smith

CIRCULATION DEPARTMENT

subscribe@servomagazine.com

MARKETING COORDINATOR

WEBSTORE

Brian Kirkpatrick

sales@servomagazine.com

WEB CONTENT

Michael Kaudze

website@servomagazine.com

ADMINISTRATIVE ASSISTANT

Debbie Stauffacher

PRODUCTION

Sean Lemieux

Copyright 2013 by

T & L Publications, Inc.

All Rights Reserved

All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *SERVO Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *SERVO*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *SERVO*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: **430 Princeland Court, Corona, CA 92879.**

Printed in the USA on SFI & FSC stock.



The sky is not the limit.



With the right parts your robot can go anywhere.



www.solutions-cubed.com
designservices@solutions-cubed.com
530.891.8045

Solutions³

Solutions Cubed... hardware made easy

for an additional 5% discount use coupon code SERVOMAG13 (through 1/1/14)

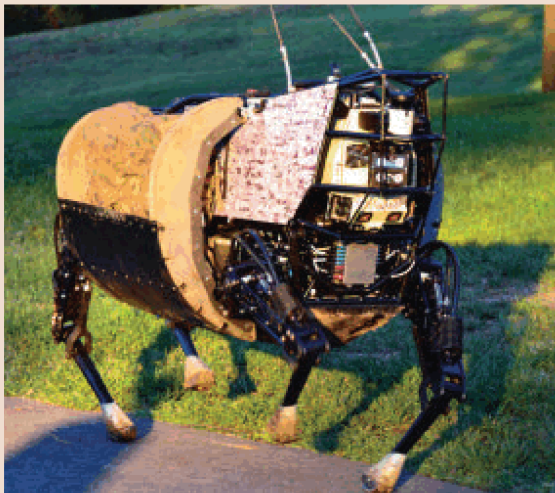


Beast of Burden Aces Field Tests

Back in January 2010, the Defense Advanced Research Projects Agency (DARPA) sent Boston Dynamics (www.bostondynamics.com) \$32 million to put together an advanced bot, dubbed the Legged Squad Support System (LS3). The aim was to create a highly mobile, semi-autonomous legged robot that can carry a 400 lb load, follow squad members through rugged terrain, and interact with troops in the same way a trained animal works with its handler.

Right on schedule, the mechanical mule prototypes recently were put through two weeks of field tests in the woods around Ft. Pickett, VA. Working with folks from the Marine Corps Warfighting Lab, researchers "demonstrated new advances in the robot's control, stability, and maneuverability, including Leader Follow decision making, enhanced roll recovery, exact foot placement over rough terrain, the ability to maneuver in an urban environment, and verbal command capability." In other words, this thing can walk around trees, step over logs, and navigate through rough terrain of various sorts.

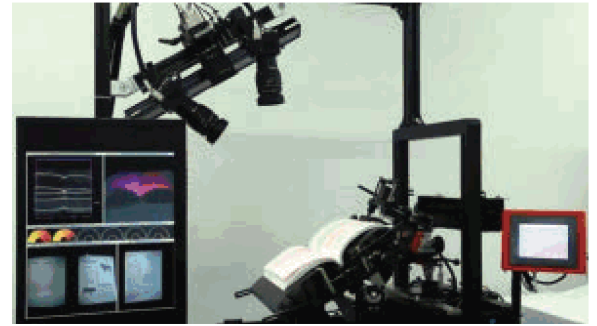
The contraption is pretty noisy, so it's not likely to be a good companion when trying to sneak up on the enemy, but it will carry a load that no backpacked soldier could ever handle. The best way to comprehend what this thing can do is to watch the vid (search for "LS3 Follow Tight" on YouTube), in which it does all these things and even recovers itself after rolling over into a dry creek bed.



Tests and demonstrations are slated to continue through the first half of next year.



The amazingly bovine LS3 traverses woods, steep inclines, ditches, and other rough terrain.



Bot scans and processes 250 ppm.

High Speed Book Scanner

Those of us who have had occasion to scan and digitize books are painfully aware of how much time it takes to strip off the jacket and separate the pages, load a page onto a flatbed scanner, scan it, and repeat ad infinitum. Sure, stack feeders can improve the process, but they are prone to misfeeds and can't do anything about the snail-like scan rate.

Not surprisingly, however, robotics technology has found a way to spare us this drudgery via the BFS-Auto book scanner, recently developed at the University of Tokyo's Ishikawa Oku Laboratory (www.k2.t.u-tokyo.ac.jp).

Perhaps the most interesting feature is that it does not require destruction of a scanned book — it just flips through the pages one at a time. The software analyzes the curvature of each page at a rate of 500 times per second, automatically compensates for image deformation, and restores the original image. Even at 400 ppi resolution, the device can digitize up to 250 pages within a minute.

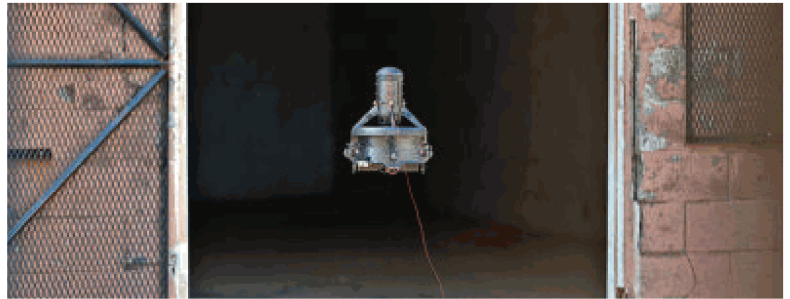
No price tag was announced, and it may be that the market for such a device is highly limited — especially given that commercial outfits already exist that digitize books for about a penny a page. For high volume users like libraries and universities, though, this might be just the thing.

Hovering Snoop Offers Endurance, Security

There is no shortage of small, flying VTOL designs these days, but the Extreme Access System for Entry (EASE) from CyPhy Works (cyphyworks.com) is a bit different. Instead of carrying its own battery power, it is linked to a lightweight ground control station via a very thin "microfilament" tether that provides both power and communication to the vehicle.

This design allows you to fly it almost indefinitely by performing battery hot-swaps on the ground. Plus, it offers more secure operation, as its communications cannot be intercepted, jammed, or spoofed.

The microfilament — which spools out from the vehicle — is said to be unaffected by water, power line EMI, and other hazards. In addition, "the microfilament enables high quality, full-frame-rate unbroken HD video



CyPhy's EASE operates in close quarters, beyond line of sight, and in GPS denied areas.

that no other small or micro UAS can match."

Features include 300 ft AGL maximum altitude and a 1,000 ft range with autonomous launch, hover, and land, plus down-looking and forward-looking HD color video and optional thermal cameras. The 12 x 16 in device operates up to 50 min between battery swaps.

Fiddler on the Beach

Those of us who grew up around salt marshes and mangrove swamps have long been familiar with the comical little creatures called fiddler crabs. Defying the principle of symmetry in nature, the males of the species sport a single oversized claw, which one might assume to be somehow related to self-defense.

According to an Australian National University (www.anu.edu.au) study conducted with the assistance of a set of "RoboCrabs" with varied claw sizes, its main function is actually to attract females, and the bigger, the better. In a fairly complicated ritual, the males display their appendages and wave them around in an effort to impress the ladies. (This sort of thing is not uncommon in nature and, in fact, a couple of my college buddies were taken into custody for similar behavior.)

The study indicates, however, that a female is not so easy to woo and, in fact, she may scrutinize as many as 100 males and their burrows before deciding that he is Mr. Right. In addition, "We know that females prefer larger claws and faster waving," according to Ph.D. student Sophia Callander, who has been studying the subject for three years. "They also prefer leaders: males who wave slightly before the other members of the courtship group."

In spite of their fussiness, females seem incapable of telling the difference between a real crab and a RoboCrab,

which consists only of a plaster claw attached to a moveable metal arm. The bottom line is that successful males tend to (a) have the largest claws; and (b) cleverly surround themselves with smaller-clawed males to make themselves look more manly.

The moral: If you're going bar hopping and trolling for women, take a few nerdy friends with you for comparison. **SV**



A (real) fiddler crab attempts to attract females.



GEER HEAD

by David Geer

Contact the author at geercom@windstream.net

Discuss this article in the
SERVO Magazine forums at
<http://forum.servomagazine.com>

HDT Global's Adroit MK2 Robot Arms

They're no Transformers, but they produce a lot of action!

The MK series of robotic arms from HDT Global's Robotics division may not be able to change into Dodge Chargers, 18-wheelers, or military aircraft, but they are able to do a lot of digging, gripping, chores with power tools, path clearing, and bomb removals.

The HDT Global Adroit MK2 robotic manipulator arm and end effector attachments were originally conceived and developed as part of DARPA's Revolutionizing Prosthetics program. The impetus was to create an altogether new type of arm prosthesis. The result is a robot arm that performs practical real world tasks with more than human levels of strength, saving lives and the cost of inserting people in complex working spaces.



The MK1

HDT's MK series of robotic arms may be no match for the Transformer's Megatron, but it is a step closer to more strength, manipulation, and increased degrees of freedom in controlled practical robotic arm and end effector technologies.

As first in a series of military robots, MK1 was used primarily for explosive ordnance disposal (EOD) to remove bombs while protecting military personnel and civilians, keeping them at a safe distance.

The MK1 has near-human arm and hand manipulation capabilities, and better than human levels of strength and durability with the capacity to carry 50+ lb with its own total mass being only 14.5 lb.

Applied to both military and homeland security needs, the MK1 has been exceeding the load-to-mass and load-to-volume ratios of concurrent similar technologies in the field and was one of the most technologically advanced robot arms used in defense applications.

The dexterity and control of the MK1 are state-of-the-art, based on 11 actuated degrees of freedom and 17 articulated joints. Its arm and wrist are comprised of five modules with seven degrees of freedom. The MK1's robot hand has four actuators that move 10 articulated joints. The hand has a grasp and manipulation capability beyond any concurrent field operational device.

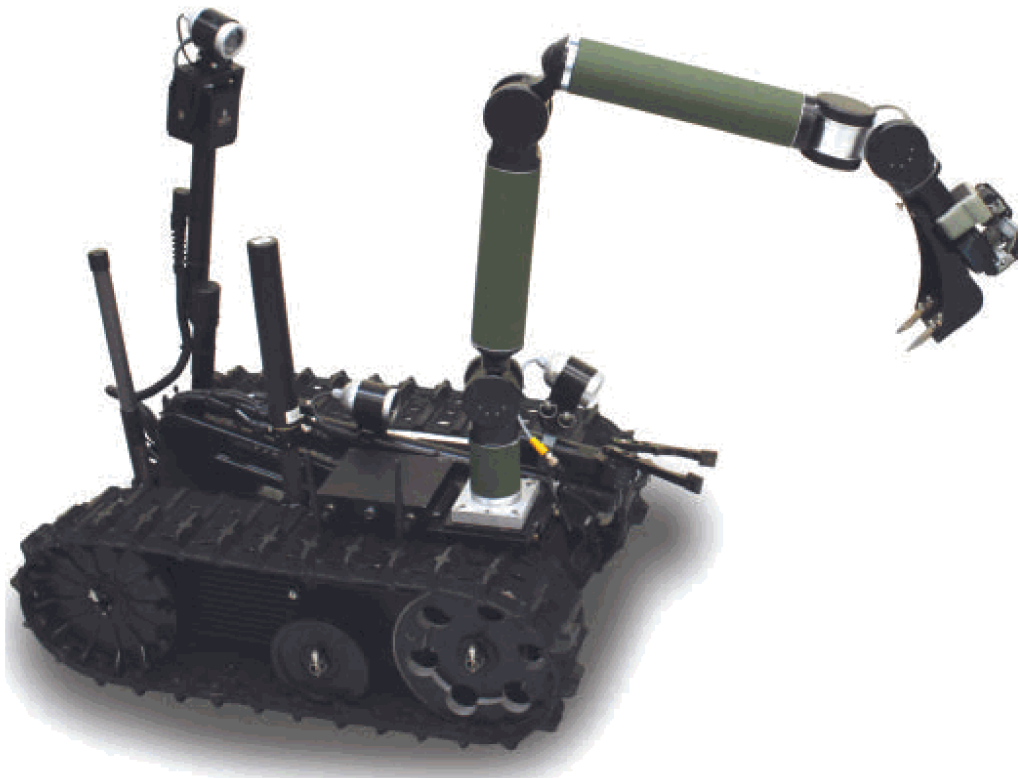
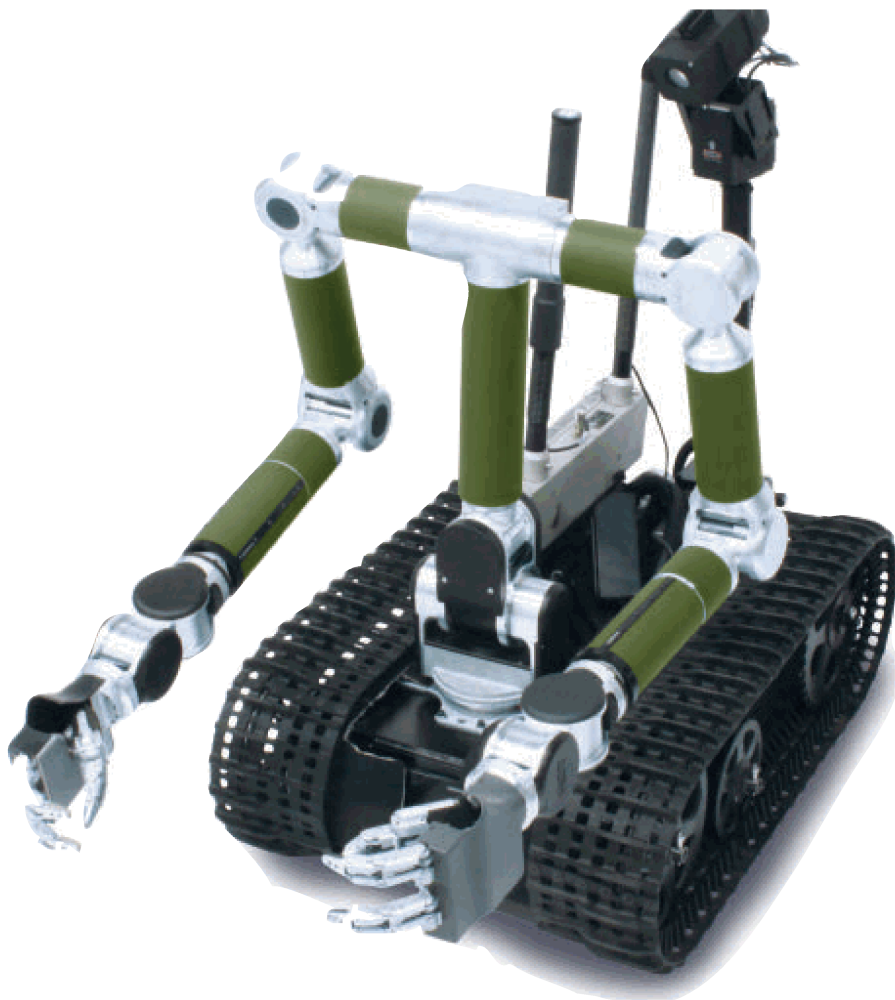
The MK2

HDT Global's Adroit MK2 arm is the next generation in their dexterous manipulator family (including single arm and dual arm manipulation capabilities, and left- and right-handed configurations) and is a thorough update to the MK1. The MK2 has the same load capacity and weighs just 16 lb. It is ruggedized and watertight, so stays sealed against intrusion from water, dust, and dirt.

The MK2 is lighter and three times faster than other robot arms. The three-finger conformal end effector has four degrees of freedom with an opposable thumb, so can use a variety of tools designed for a human hand. The MK2 also includes mounting rails to incorporate a variety of accessories, including a bucket that allows the arm to be used as a miniature backhoe. These capabilities are enabled in part by a torque sensor and absolute position sensor.

The MK2 robotic arm's modular components use aluminum and carbon fiber as their primary structural materials and components.

"High strength steel is used in very high stress areas such as gear teeth and the modular arm connectors. We use brushless DC motors which have permanent



magnets and copper windings," says Dr. Tom Van Doren, Chief Operating Officer at HDT Global's Robotics division.

The modules are attached via uniform connectors, each with its own power and communications including both a CAN and Ethernet bus. This enables the arm to gather video if a camera is attached, as well as to aid in manipulation.

The MK2 robotic arm is currently compatible with the iRobot Packbot and the QinetiQ TALON robot via retrofit mounting kits. "These kits require no changes to the robot (including software), and allow the Adroit arm to be used together with the original arm on the robot. We also have a general-purpose mounting kit

Resources

HDT Global, maker of the Adroit MK2 robotic arm and attachments
www.hdtglobal.com

The HDT Global Adroit MK2 robotic arm in action at the AUVSI Unmanned Systems Show in 2012
www.engineeringtv.com/video/We-Hand-Over-the-Microphone-to

that allows the arm to be mounted on any flat surface that is about 4" by 4". We supply wireless links to operate the arms," explained Dr. Van Doren. Other retrofit kits for popular robots are now in development.

In the Field

The Adroit MK2 robotic arms are in use today by several military organizations, NASA, and commercial customers. They are being used for laboratory and field use for a range of applications, and are continuing to be evaluated for use on EOD robots. A marine variant of the arm that can operate to depths of 100 meters in salt water is in design now and will be built in 2013.

As with other robots, the MK2 manipulator arm and its attachments can perform tasks that normally would put people at risk. "We think of the Adroit arms in general as a way to provide human dexterity remotely. Our key distinguishing characteristics are very high dexterity (we have systems with up to 27 degrees of freedom), plus a

very high payload-to-weight ratio. Thus, the arms can be mounted on very lightweight systems and provide high dexterity and high lift capacity," Dr. Van Doren continued.

The arm is controlled by a human operator using a complex joystick, which was built in-house. The operator's hand rests in the joystick which reads the movements of the operator's hand, two fingers, thumb, and wrist, and then translates those to the robot arm and hand.

The MK2 can also be used in places where it would be too expensive to send a human being. For example, the Adroit arms can be used for repetitive tasks that require high dexterity and light weight, according to Dr. Van Doren.

While the Adroit MK2 arm is currently in use in research projects at universities and in private industry, journal articles about research results have not as yet been published.

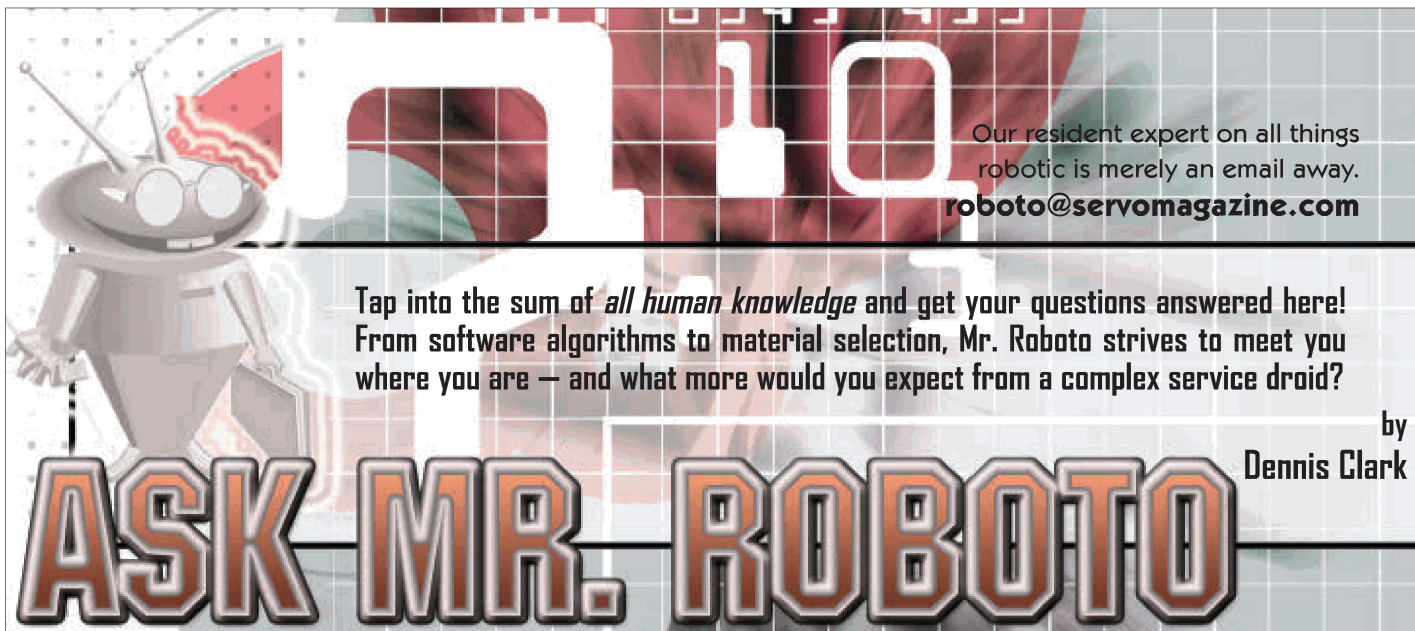
Conclusion

The MK series of robotic arms offers advanced capabilities that have not been previously available. As robots get closer to matching human dexterity, there will be even more ways to integrate robots into our lives. While HDT Global did not mention any plans to attach MK series arms to humanoid robots in the foreseeable future, it is working on healthcare applications and continues to be available for domestic law enforcement tasks. **SV**



Protector Series

HDT Global's Robotics division is gearing up to release its own Protector series of logistics and trail clearance robots, which will also be compatible with the MK2 arms. The Protector robots will be powered by diesel — or JP8 — and will haul a load up to 1K lbs. The Protector will have more than three days of mission level operational capacity using existing onboard fuel.



Correction from the January Column

First off, I need to start out with a correction. In **Figure 6** back in the January 2013 issue, I mislabeled the Roomba serial control mini DIN connector. The pin that is marked as pin 6 is actually pin 7. **Table 2** at the end of that same article which lists the functions of the pins is correct.

Fortunately, both of these pins are the Roomba's battery minus pins, so no matter how you connected them, there wouldn't be any problems.

Now, back to our Roomba control round-up ...

More on the Hardware

Speaking of Roomba power connectors, you can't just plug your Arduino into it without some preparation. The output of the Roomba battery can run as high as 20V while charging; while it is running, it is over 17V.

If you plug that power into the input jack of your Arduino, you will overheat your regulator and probably blow one of the voltage comparator chips! You need to step this voltage down to use it.

Spring is just around the corner, but winter doesn't seem ready to let go yet. There is still plenty of time to spend indoors with our robots, preparing for the next competition. That is certainly what I'm doing. The 2013 Critter Crunch is coming up in just a few months! (Ahh!) For this column, I'm going to do some more with Roomba controlling with an Arduino UNO in the hopes that this project will boost me well on my way to a new 20 pound Critter to compete with those other critters that still use "wetware" as the controller! Which, I have to admit, is pretty much all of them. Eventually, my robots will win the majority of its combats, and then win them all! Bwahahaha! Ahem.

Anyway ... This bit of Roomba controlling is a bit late, but better late than never (someone once said ... does anyone actually know who first said that?). Anyway, on with the show!

Here, you have two choices: You can either build a power board using a 7812 style voltage regulator and plug that 12V into your Arduino power input jack (which handles up to 12V safely); or you can step the voltage down to 5V and plug that into your Arduino 5V connector.

I didn't have any 12V linear regulators, but I did have some Power Trends PT78ST105 5V switching regulators. These will take from 7V to 27V input and provide 1.5 amps of 5V as output. Just add two capacitors and you're done!

Figure 1 shows my setup with the 5V regulator board and Arduino UNO board sitting on top of my "FrankenRoomba" robot. The one main problem with this setup is that power is *always* available at the SCI

connector so the Arduino comes on and wakes up the Roomba as soon as it is plugged in.

There is a two second wait at the beginning of the program that gives you time to put it down and move away before it takes off.

How the Firmware Works

Take a look at the source code listing and use that as a reference to my following explanation of the very simple code: *RoombaScoot*. You can grab the zip file from the article link. You must use at least Arduino 1.0 or later for this program.

Let's break the code down by sections and discuss what is going on.

LINEAR SERVOS

L12-R Linear Servo

- Direct replacement for regular rotary servos
- Standard 3 wire connectors
- Compatible with most R/C receivers
- 1-2ms PWM control signal, 6v power
- 1", 2" and 4" strokes
- 3-10 lbs. force range
- 1/4" to 1" per second speed ranges
- Compatible with VEX

L16 Linear Actuators

- 2", 4" and 6" strokes
- 10 - 40 lbs. force range
- 1/2" to 1" per second speed ranges
- Options include Limit Switches and Position Feedback

New!

PQ12 Linear Actuator

- Miniature Linear Motion Devices
- 6 or 12 volts, 3/4" stroke
- Up to 5 lbs. force
- Integrated position feedback or limit switches at end of stroke
- External position control available

Linear Actuator Controller (LAC)

- Will drive any Linear Actuator with position feedback
- Up to 24v and 4 Amps
- USB connectivity to drive the actuator with your computer
- Adjustable speed, limits and sensitivity

L12-NXT Linear Servo

- Designed for LEGO Mindstorms NXT®
- Plugs directly into your NXT Brick
- NXT-G Block available for download
- Can be used with Technic and PF
- Max. speed: 1/2" per sec.
- Pushes up to 5 lbs.
- 2" and 4" strokes

Available Now @
www.firgelli.com

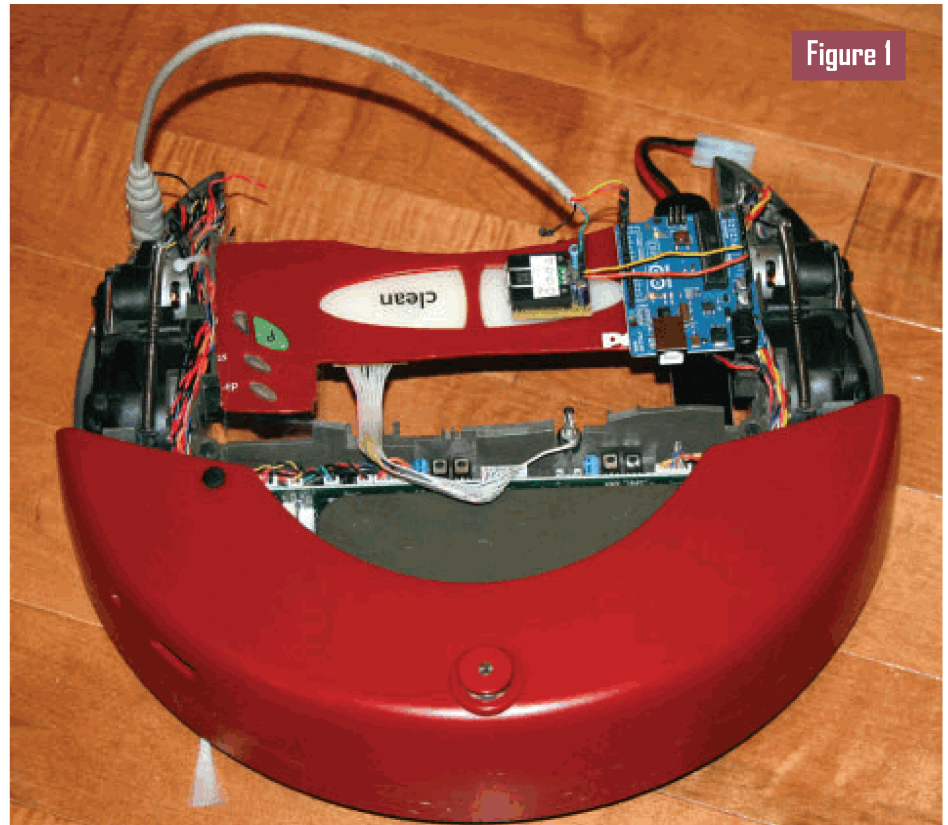


Figure 1

Definitions

At the top of the program are all of the includes, object defines, and macros that are used. The first version of this program used the Arduino hardware serial port to control the Roomba. Some people wrote about problems between the Arduino and the Roomba with this solution. It also didn't allow any kind of debugging information to be sent back to the computer.

The next version of the code that came just a few brief months later (way back in 2006) used the hardware serial port to talk back to the debug monitor of the Arduino IDE and a third party library called *NewSoftSerial* (which I wrote about a few years ago) to control the Roomba.

This worked better and allowed debug data to be returned to the computer. With a wireless transceiver attached, it allowed wireless remote control, as well.

Time passed and the Arduino IDE changed and improved. One of those improvements was to implement the *NewSoftSerial* type software serial

port into the regular libraries of Arduino 1.0 and later. However, the syntax changed a lot with the 1.0 release. My rewrite — the third rendition of this program — uses the most recent Arduino IDE and adds new functionality and performance improvements (if I do say so myself).

Moving along, the software serial port uses the Arduino *SoftwareSerial* class; its ".h" file is near the top. Shortly after that, we define the pins we are going to use in our program and then instantiate the *SoftwareSerial* object. There are two macros defined which look at bits 0 and 1 of the first byte in the sensor data array, called *bumpwright* and *bumpleft*. You'll see how those are used later on.

Setup

The Arduino has two main sections: **Setup** and **Loop**. This is pretty much how all embedded firmware works when you get right down to it. Here, we define the baud rate of both the *mySerial* port (software serial to the Roomba SCI

port) and the hardware port. **Serial** is always defined, so all we need to do is set its baud rate.

Next, we want to wake up the Roomba. To do that, we pulse the **DD** line low for 500 ms. You should hear your Roomba beep and its lights will come on. As soon as it wakes up, a bunch of "stuff" is spewed out. I found that this only occurs if your Roomba is still in its *Vacuum* mind.

As soon as I successfully got my Roomba into *Control* mode, it stayed there and never spit that data out again. No big deal, it wasn't that interesting anyway. It just told how long the Roomba had been asleep.

After the Roomba is awake, we want to put it into one of its control modes. The sequence detailed at the end of the **Setup** section is what SCI requires to send commands to a Roomba. This sequence moves the Roomba from *Passive Mode* to *Safe Mode*. Check pages 3 and 4 of the *iRobot Roomba Serial Command Interface (SCI) Specification* that is available at www.irobot.com.

Safe Mode is good for us because no matter what commands we give our Roomba, the safety cutoff for the motors remains active. This will help protect our robot from driving off the stairs! There are other commands that move Roomba to *Full* and back to *Safe* modes listed, as well. We'll stay "safe."

Loop

This is the main "bump-n-go" loop for our robot. The progress is simple:

```
Loop
  Look at sensors
  If bumped into something on the
  left then:
    Backup and turn right
  Else if bumped into something on the
  right then:
    Backup and turn left
  Else
    Go forward
End Loop
```

I added a little "eye candy" by turning on the corner sweep brush motor whenever a bump action is active.

Let's look at the basic functions to see how they work. I've left these command sequences written out long hand instead of just packing the data into an array and spitting it out so that you can see all the commands and data easily. Normally, I'd write a single function to do the command sequence writing and put all the commands and data into a byte array for execution.

Most of the commands in this program are just telling the Roomba to move. Page 4 of the SCI manual details the *Drive* command. You tell the Roomba how to move by using the "137" command and then sending two 16-bit signed integers.

The first 16-bit integer defines the speed to move. The

range is -500 to 500 millimeters/second.

The second 16-bit integer is the direction. This can be a number from -2000 to +2000, and defines the millimeter radius you want the robot to turn. Negative numbers turn right; positive numbers turn left. The larger the number, the smaller the turn. There are three special numbers:

Straight	32768 (0x8000 hex)
Pivot Clockwise	-1
Pivot Counterclockwise	1

I found that my Roomba will not drive straight; it always turns to the right. I think that I need to clean the motor rotational sensors in the motor hubs. At least one of them is not properly reporting wheel position.

The other half of the Roomba controls is the *UpdateSensors()* function. This program is set to get **Group 1** sensor data (see page 7 of the SCI manual). These bytes give us the status of the front bumper switches, wheel drop switches, "cliff" sensors, virtual walls, motor over-current, and dirt detectors.

We are only looking at the first byte which is the front bumper switches. The two *#define* macros look at bits 0 and 1 which are those bumper switches. If we don't get data back, we have an LED flasher loop that tells us.

That's all there is to it! Then, why did I have so much

FastProtos.com

3D PRINTING SERVICE

**Custom Parts
in 1-2 Days!
\$1+ per part**



**- Precise, Smooth Plastic
- Upload SketchUp or CAD
at FastProtos.com!**

trouble? My first problems came when my Roomba was not at the baud rate it was supposed to be. For some reason, it was at 19200 baud instead of 57600. Why? I'm not sure.

I read in one place that you could change that baud rate by a magic handshake involving holding the "ON" button for a certain amount of time. Regardless, there is a command to change the baud rate which I did in a one-

time program. Then, I found that in Safe mode, the motors don't work. So, when I have the Roomba suspended on my desk it wouldn't take off ... it *REALLY* wouldn't take off!

I also spent some wasted time trying to confirm if I had the right firmware version on my Roomba (later than October 2005). iRobot would not tell me anything about "anything that you can't find on our site yourself" and

Listing 1: RoombaScoot sketch.

```

/*
 * RoombaBumpTurn 3.0
 * -----
 * Implement the RoombaComm BumpTurn program
 * in Arduino
 * A simple algorithm that allows the Roomba
 * to drive around and avoid obstacles.
 *
 * Arduino pin 4 (RX) is connected to Roomba
 * TXD
 * Arduino pin 3 (TX) is connected to Roomba
 * RXD
 * Arduino pin 2 is connected to Roomba
 * DD
 *
 * Rewritten 28 January 2013 to use Arduino 1.0
 * SoftSerial, act a little differently and use
 * the hardware serial port for debug
 * transmission. (Dennis Clark)
 *
 * Updated 20 November 2006 with the idea of
 * using two serial ports, one for debugging by
 * Josh Heling, from his website
 * http://www.netfluvia.org/layer8
 *
 * Originally created 1 August 2006
 * copyleft 2006 Tod E. Kurt <tod@todbot.com>
 * http://hackingroomba.com/
 */

#include <SoftwareSerial.h>

int rxPin = 4;
int txPin = 3;
int ddPin = 2;
int ledPin = 13;

/*
 * Define number of sensor bytes and the
 * byte array.
 */
#define BYTES 10
uint8_t sensorbytes[BYTES];

SoftwareSerial mySerial(rxPin,txPin);
// Roomba comm port

#define bumpright (sensorbytes[0] & 0x01)
#define bumpleft (sensorbytes[0] & 0x02)

void setup() {
  uint8_t i;

  pinMode(ddPin, OUTPUT);
  // DD "wakeup" pin is an output
  pinMode(ledPin, OUTPUT);
  // LED pin is an output

  /*
   * Set up the software serial port to talk to
   * the Roomba.
   */
  mySerial.begin(57600);

  /*
   * Set up the hardware serial port for
   * debugging. Most of the debug output is
   * commented out so that it doesn't
   * slow down the response times reacting to
   * bumper hits.
   * To use them, uncomment the "Serial.print()"
   * methods.
   */
  Serial.begin(57600);

  Serial.println("Wake up Roomba\n");
  digitalWrite(ledPin, HIGH);
  // say we're alive
  digitalWrite(ddPin, HIGH);
  delay(100);
  digitalWrite(ddPin, LOW);
  delay(500);
  digitalWrite(ddPin, HIGH);

  /*
   * Grab whatever interesting stuff the Roomba
   * transmits and send it to debug port.
   */
  while(mySerial.available())
  {
    i = mySerial.read();
    Serial.write(i);
  }
  Serial.println();
  delay(2000);

  /*
   * Set up Roomba to receive commands in
   * "Safe" mode.
   */
  Serial.println("set up ROI");
  mySerial.write((uint8_t)128); // START
  delay(50);
  mySerial.write((uint8_t)130); // CONTROL
  delay(50);
  digitalWrite(ledPin, LOW);
  // say we've finished setup
}

/*
 * This main loop is really simple. At the top
 * of each pass through the loop ask for Sensor
 * Packet 1 (10 bytes). We are only looking at
 * the first byte which gives the state of the
 * bump sensors.
 * If we see a bumper response on the left or
 * the right, act accordingly, otherwise just
 * go forward.
 * Additionally, if any bumper is seen, we'll

```

referred me to the Arduino forums for the Roomba here: www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1185405709. There is a lot of information built up in there with six years worth of hacking!

Well, that's all folks. I hope this makes it much easier for you to play at controlling your Roomba with your

favorite microcontroller. I've got a list of things to try like remote control and, well, lots of stuff. I have to clean my rotation sensors first though ...

Remember, if you have a robot question, please drop me an email at roboto@servomagazine.com and I'll do my best to answer them. Until next month, keep on building robots! **SV**

```
* turn on the corner brush motor so we can
* tell what is happening.
*/
void loop() {
  digitalWrite(ledPin, HIGH);
  // say we're starting loop
  UpdateSensors();
  //Serial.print("Sensor = ");
  //Serial.println(sensorbytes[0],HEX);
  //Serial.println("");
  //delay(1000);

  if(bumpleft) {
    DoBrush(1);
    GoBackward();
    delay(1000);
    SpinRight();
    delay(1000);
    DoBrush(0);
  }
  else if(bumpright) {
    DoBrush(1);
    GoBackward();
    delay(1000);
    SpinLeft();
    delay(1000);
    DoBrush(0);
  }
  GoForward();

  digitalWrite(ledPin, LOW);
  // We've ended a loop
}

void DoBrush(uint8_t on) {
  // Turn side brush on or off
  if (on)
  {
    mySerial.write((uint8_t)138);
    mySerial.write((uint8_t)0x01);
  }
  else
  {
    mySerial.write((uint8_t)138);
    mySerial.write((uint8_t)0x00);
  }
}

void GoForward() {
  mySerial.write((uint8_t)137);
  // DRIVE
  mySerial.write((uint8_t)0x00);
  // 0x00c8 == 200
  mySerial.write((uint8_t)0xc8);
  mySerial.write((uint8_t)0x80);
  mySerial.write((uint8_t)0x00);
}

void GoBackward() {
  mySerial.write((uint8_t)137);
  // DRIVE
  mySerial.write((uint8_t)0xff);
  // 0xff38 == -200
  mySerial.write((uint8_t)0x38);
  mySerial.write((uint8_t)0x80);
  mySerial.write((uint8_t)0x00);
}

void SpinLeft() {
  mySerial.write((uint8_t)137);
  // DRIVE
  mySerial.write((uint8_t)0x00);
  // 0x00c8 == 200
  mySerial.write((uint8_t)0xc8);
  mySerial.write((uint8_t)0x00);
  mySerial.write((uint8_t)0x01);
  // 0x0001 == spin left
}

void SpinRight() {
  mySerial.write((uint8_t)137);
  // DRIVE
  mySerial.write((uint8_t)0x00);
  // 0x00c8 == 200
  mySerial.write((uint8_t)0xc8);
  mySerial.write((uint8_t)0xff);
  mySerial.write((uint8_t)0xff);
  // 0xffff == -1 == spin right
}

void UpdateSensors() {
  int8_t i = 0;
  mySerial.write((uint8_t)142);
  mySerial.write((uint8_t)1);
  // sensor packet 1, 10 bytes

  while(!mySerial.available());
  // wait for data to appear.

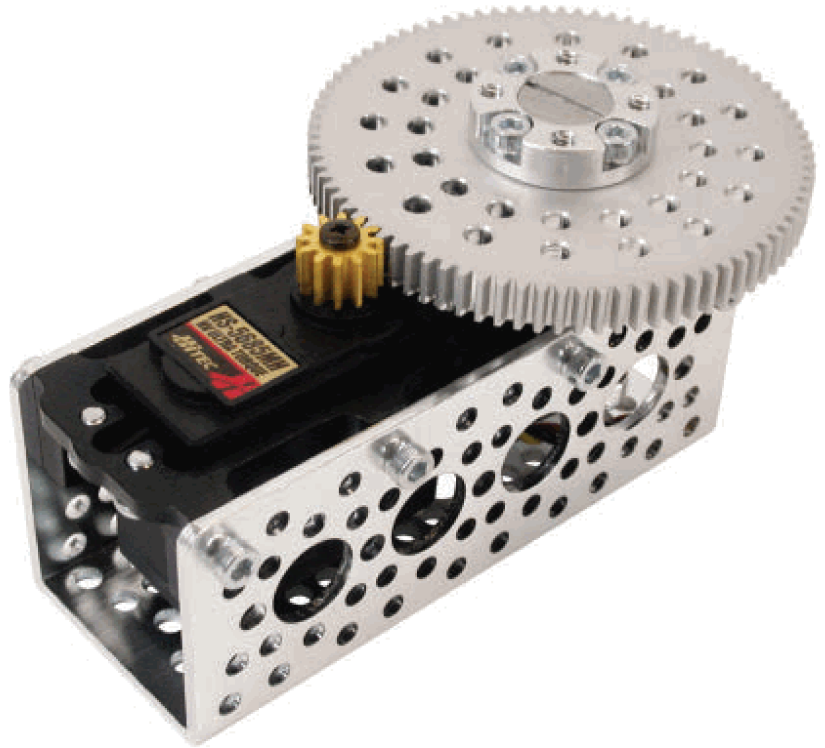
  while(mySerial.available())
  // get 10 bytes of sensor data
  && (i < BYTES) )
  {
    int c = mySerial.read();
    if( c== -1 ) {
      for( int i=0; i<5; i++ ) {
        // say we had an error via the LED
        digitalWrite(ledPin, HIGH);
        delay(50);
        digitalWrite(ledPin, LOW);
        delay(50);
      }
    }
    sensorbytes[i++] = c;
  }
  //Serial.print("Got
  ");Serial.print(i);Serial.println(" bytes");
  //i = 0;
  //while (i++ < BYTES)
  //{
  //  Serial.print(i,HEX); Serial.print(" ");
  //}
  //Serial.println();
}
```


NEW PRODUCTS

Channel Mount Servo Power Gearboxes

The first in a new line of gearbox products to be released by ServoCity this year are the new channel mount servo power gearboxes. By incorporating threaded side mounts, users can easily attach these servo gearboxes to numerous ServoCity products, making it easier to incorporate a servo power gearbox into a custom pan & tilt, mechanical device, or robotic structure.

Full metal gears (32 pitch) provide strength and durability. By utilizing digital servo technology, the user is able to tailor the gearbox to fit various performance parameters. The .770" pattern on the output hub gives nearly unlimited attachment possibilities. You can select from 90°, 180°, 360°, or continuous rotation.



90° Aluminum Tube Clamps

Also from ServoCity are new tube clamps for attaching two pieces of shafting or tubing at a 90° angle. Constructed from 6061-T6 aluminum for superior strength, the clamps don't easily flex or bow.

The 5/8" bore 90° tube clamp is perfect for builds that require smaller tubing in tight spaces. The 5/8" bore version comes with two 6-32 x 3/8" socket head screws. The larger 90° tube clamp with a 1" bore is designed for bigger projects that require 1" tubing or shafting. The 1" bore clamp also has two additional 6-32 thru holes at the bend that provide an additional mounting point to other components. The larger tube clamp includes two 6-32 x 1/2" socket head screws.

The 5/8" bore tube clamp is \$7.99; the 1" bore tube clamp is \$8.99.

For further information, please contact:

ServoCity

Website: www.servocity.com

Is your product innovative, less expensive, more functional, or just plain cool? If you have a new product that you would like us to run in our *New Products* section, please email a short description (300-500 words) and a photo of your product to:

newproducts@servomagazine.com

New Board For TIMSP430F5172

SchmartBoard has expanded its product offering with a new development board to support users of Texas Instruments microcontrollers. These boards use their "EZ" soldering technology to assure fast, easy, and flawless soldering.

The MSP430F51x2 series are microcontroller configurations with two 16-bit high resolution timers, universal serial communication interfaces (USCI_A0 and USCI_B0), a 32-bit hardware multiplier, a high performance 10-bit analog-to-digital (A/D) converter, on-chip comparator, three-channel DMA, 5V tolerant I/Os, and up to 29 I/O pins. The MSP430F51x1 series are microcontroller configurations with two 16-bit high resolution timers, universal serial communication interfaces (USCI_A0 and USCI_B0), a 32-bit hardware multiplier, on-chip comparator, three-channel DMA, 5V tolerant I/Os, and up to 29 I/O pins.

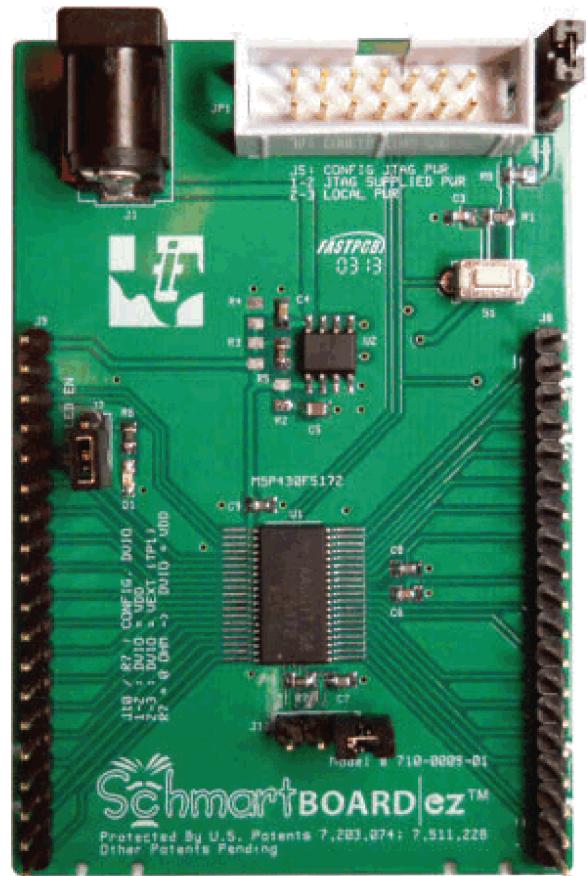
Typical applications for these devices include analog and digital sensor systems, LED lighting, digital power supply, motor control, remote controls, thermostats, digital timers, handheld meters, etc.

The board is priced at \$35.

For further information, please contact:

Schmartboard

Website: www.schmartboard.com



Transparent Proto-Boards

Global Specialties has introduced two new and unique transparent Proto-Boards® — models PB-242T and PB-326T.

These two new transparent models have ABS plastic sockets with phosphor-bronze contacts that are mounted on metal back plates for added durability. The transparent ABS plastic sockets are perfect for educational applications because they allow students and designers to clearly view circuit connections from the front of the Proto-Board. Each of these new models offer plastic feet, color-coded and marked contact areas, and four binding posts (three red and one black).

The Proto-Boards can be used to design test fixtures, perform lab experiments, and other areas of custom electronic design applications. The new transparent model PB-242T with 2390 tie points lists for \$65; the PB-326T with 3,220 tie points lists for \$89.



Microprocessor Design Trainer

Also available from Global Specialties is the DL-030 Microprocessor Design Trainer. The DL-030 utilizes an FPGA chip allowing a user to implement a fully functioning microcontroller in minutes. The DL-030 is a complete and ruggedly packed trainer that will provide a solid learning platform for individuals and classroom students.

With the DL-030, you can create and implement your first design in as little as one hour. Features include:



Continued on page 42

EVENTS

Calendar

ROBOTS.NET

Send updates, new listings, corrections, complaints, and suggestions to: steve@ncc.com or FAX 972-404-0269

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: <http://robots.net/rcfaq.html>.

— R. Steven Rainwater

MARCH

8-9 **AMD Jerry Sanders Creative Design Contest**
University of Illinois at Urbana-Champaign, IL
Autonomous robots built by college teams compete in one of the oldest university level robot competitions.
<http://jsdc.ec.illinois.edu>

8-10 **Cognizance**
IIT Roorkee, India
A variety of events for autonomous robots.
<http://cognizance.org.in>

9-10 **METU Robotics Days**
METU Culture and Convention Center, Turkey
Robots compete in events that include line following, Sumo, mini Sumo, multi-mini Sumo, minefield, triathlon, and a new search and rescue contest.
<http://topluluk.odturobotgunleri.org.tr>

14-16 **Festival de Robotique**
Stade Uniprix, Montreal, Quebec, Canada
FIRST events include FRC, FLL, and jrFLL.
<http://festivalderobotique.ca>

14-17

Techkriti RoboGames

Indian Institute of Technology, Kanpur, Uttar Pradesh, India
Multiple events for autonomous robots.
www.techkriti.org

15-19

Apogee iStrike

Zuarinagar, Goa, India
Check the website for forthcoming details on this year's robot events.
www.bits-apogee.org

16

Manitoba Robot Games

Tec Voc High School, Winnipeg, Manitoba, Canada
Events include Sumo, mini Sumo, mini tractor pull, Super Scramble, line following, and Robo-Critters.
www.scmb.mb.ca

17-21

APEC Micromouse Contest

Long Beach, CA
Super-fast 25 cm square micromouse robots compete in maze solving.
www.apec-conf.org

23-24

RobotChallenge

Vienna, Austria
Lots of events for autonomous robots including standard Sumo, mini Sumo, micro Sumo, the Parallel Slalom, and the Slalom Enhanced.
www.robotchallenge.org

APRIL

1

Istrobot

STU, Bratislava, Slovakia, EU
Events include Micromouse, line following, mini Sumo, and free style.
www.robotics.sk

4-10

Alcabot-Hispabot

University of Alcalá, Madrid, Spain
Sumo, mini Sumo, Robocup Jr, Eurobot, Rastreadores, and Velocistas.
www.depeca.uah.es/alcabot

6-7 **Trinity College Fire Fighting Home Robot Contest**
Trinity College, Hartford, CT
 Autonomous robots must find and extinguish a fire.
www.trincoll.edu/events/robots

9-11 **DTU RoboCup**
Copenhagen, Denmark
 Autonomous robots compete on a course that combines line following and wall following.
www.robocup.dtu.dk

11-13 **National Robotics Challenge**
Marion, OH
 A wide range of events for student-built robots including Sumo, mini Sumo, pick-and-place, robot construction, robot maze, rescue robot, robo hockey, Bot Ball, and VEX Sack Attack.
www.nationalroboticschallenge.org

13 **Brown IEEE Robotics Competition**
Brown University, Providence, RI
 Autonomous robots must navigate a maze.
<http://brown.edu/Departments/Engineering/Organizations/IEEE/competition>

13 **CIRC Central Illinois Bot Brawl**
Normal, IL
 Events include RC vehicle combat, Sumo, line following, line maze, and Checkers on a Sumo Ring.
<http://circ.mtco.com>

17-20 **VEX Robotics World Championship**
Anaheim, CA
 This year's contest is called VEX Sack Attack.
www.vexrobotics.com/competition

Robotics Showcase

SDP/SI
 STOCK DRIVE PRODUCTS/STERLING INSTRUMENT
www.sdp-si.com
FREE CATALOG
800.819.8900
 GEARS
 BELT & CHAIN DRIVES
 PULLEYS
 COUPLINGS
 BEARINGS
 SPROCKETS

SMALL MECHANICAL COMPONENTS

ALL ELECTRONICS CORPORATION
 Thousands of Electronic Parts Available Today!

 LEDS • CONNECTORS • RELAYS
 SOLENOIDS • FANS • ENCLOSURES
 MOTORS • WHEELS • MAGNETS
 PC BOARDS • POWER SUPPLIES
 SWITCHES • LIGHTS • BATTERIES
 and many more items...
 We have what you need for your next project.

www.allelectronics.com
 Discount Prices • Fast Shipping

Recycling & Remarketing High Technology
WEIRDSTUFF® WAREHOUSE
 Software, Computers, Electronics, Equipment, Doo-hickies
384 W. Caribbean Dr. Sunnyvale, CA 94089
 Mon-Sat: 9:30-6:00 Sun: 11:00-5:00
(408) 743-5650 Store x324
 **WE BUY AND SELL EXCESS & OBSOLETE INVENTORIES!**
FREE COMPUTER RECYCLING
 We recycle computers, monitors, and electronic equipment. M-Sat 9:30-4:00

 **GREAT DEALS!**
 Hi-tech items, electronics test equipment, and more!
GIANT AS-IS SECTION
 10,000 sq. ft. of computers, electronics, software, doo-hickies, cables, and more!

 also check out our...
eBay Store
stores.ebay.com/WeirdStuff-Inc
WWW.WEIRDSTUFF.COM

bots IN BRIEF

DEXTRE IS PUMPED!

In addition to Opportunity and Curiosity, there are some other awesome robots from NASA. For example, one that pumps gas, and one that digs dirt.

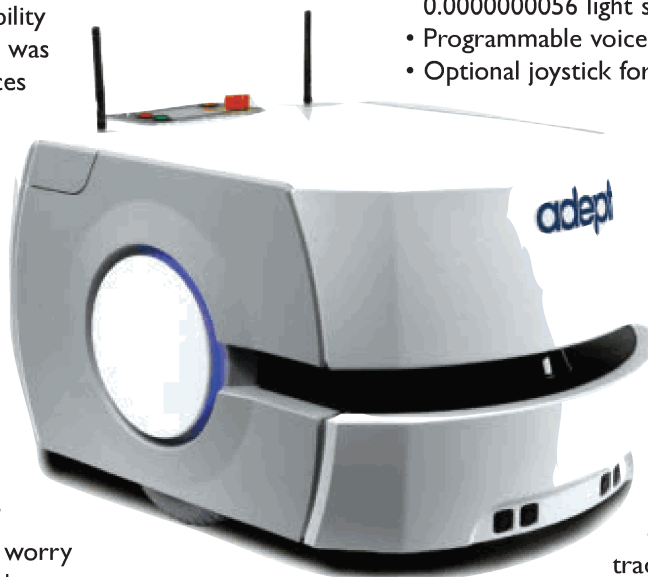
Dextre is one of these awesome bots who recently successfully completed a test refueling of a mock satellite on the exterior of the International Space Station. What's most important about this test is that the mock satellite that Dextre refueled was designed *not* to be refueled at all. With a few very pricey exceptions (like Hubble), satellites are not designed to be serviced, but Dextre showed that it could use tools to crack open normal satellites and pump fuel into them, anyway.

In this latest set of operations, Dextre removed two safety caps, cut through two sets of thin retaining wires, and finally transferred a small quantity of liquid ethanol into the washing machine sized module. The latter maneuver was particularly tricky, since handling liquids in space requires perfect precision to prevent dangerous leaks. The specialized tools built for the job allowed Dextre to seal the connections between the tool and the fuel valve to eliminate the possibility of leaks. Contributing to the level of difficulty was the fuel hose itself, which adds additional forces that tend to pull Dextre's hands. It took the combined skills of the experienced NASA and CSA robotics controllers to pull off this first-of-a-kind space refueling demonstration successfully and without any mishap.



Here's some Lynx specs:

- Weighs 60 kg, can haul 60 kg.
- 13 hour run time; 3.5 hour recharge time; autonomous recharging on dock.
- Turning radius of zero degrees thanks to differential steering.
- Maximum speed of about 4 mph, or 0.0000000056 light speed.
- Programmable voice and audio prompts.
- Optional joystick for shenanigans.



LYNX — THE “UN”ROBOT

This is Lynx — a brand new mobile robot from Adept. It's designed to move stuff from one place to another without having to worry much about what people or things may be in between.

Lynx is a self-navigating Autonomous Indoor Vehicle (AIV) designed for dynamically moving material in challenging environments that may include confined passageways, as well as dynamic and peopled locations. Unlike traditional autonomously guided vehicles (AGVs), Lynx requires no facility modifications — such as floor magnets or navigational beacons — saving users up to 15% in deployment costs. Lynx includes Adept's proprietary software and controls, allowing

it to intelligently navigate around people and unplanned obstacles that render traditional AGVs

incapacitated. Lynx can be programmed and functional within a day. Designed for developers, integrators, and end-users, the system can be customized for a variety of applications and payloads.

Adept has made a point of saying that they would not be introducing this robot as, um, well, a robot. This may seem a bit weird, but apparently in the markets that Lynx is designed to break into, robots come with a lot of perceptual baggage.

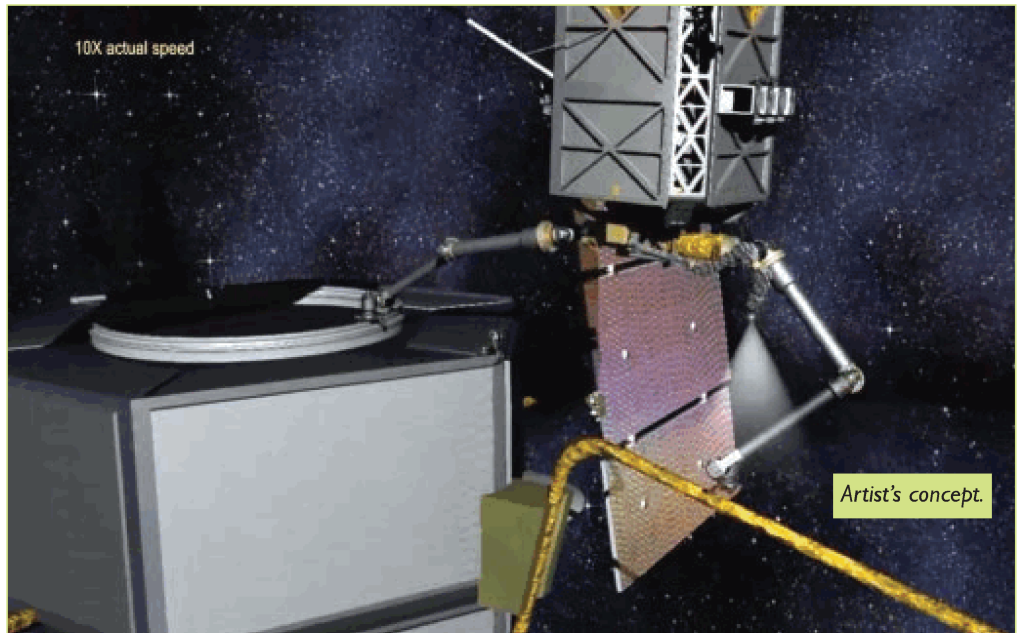
bots IN BRIEF

RISE OF THE PHOENIX

DARPA'S Phoenix program aims to create a new network of communications satellites by sending up robots to harvest body parts from old communications satellites. (Sounds like the basis for a movie plot ...)

A big part of why satellites are so expensive is that getting them from Earth into space takes rockets, and rockets don't come cheap. Plus, no matter how carefully you build hardware, sooner or later it's going to fail or go obsolete, and from that point on, your space based investment is useless to everybody. DARPA's idea is to start a sort of on-orbit recycling program, where robotic spacecraft are sent out to harvest valuable parts (like antennas) from otherwise derelict satellites, and then give them a new life by attaching new minisats to the old hardware.

The Phoenix program isn't even a year old yet, but a demonstration mission is scheduled to take place in 2016, in which a robot will remove an antenna from an old satellite and see if it can get it to do something new and useful.



"Our ultimate goal for the Phoenix program is to increase the return on investment of high value space assets by reusing components from nonfunctioning satellites that have already been placed in space through permission from their owners and techniques and technologies that allow for responsible, transparent, and safe processes and behaviors," said Dave Barnhart, DARPA program manager. "We have a long way to go, but we are laying the foundation for improving how we build space systems, with the goal of changing the economic model for space operations."

PET PROJECT

Cornell University's College of Veterinary Medicine is working on an advanced robot dog — code-named Butch — that will simulate doggy medical emergencies for veterinarians in training. The dog is part of an overall plan for a simulation center with robot cats and dogs that will be the most advanced training center of its kind in the world.

Butch will run with inexpensive, off-the-shelf electronic components and sport a more realistic airway, a soft abdomen compartment, articulating joints, more areas for catheters, more space inside the body, and a more realistic overall feel.

The new and improved robot dog will also integrate with a new veterinary simulation toolkit called Ursula (the Universal Realistic Simulation of a Living Animal) that will be open source and shared with other veterinary institutions. Ursula will simulate physiological responses for a variety of animals and will integrate with inexpensive hardware.





FACE IT, KID

A recent UCSD news release is touting their new android child that is designed to mimic the expressions of a one year old human child as it learns to control its body and interact with humans. This robot is a little less creepy and a little more life-like than other baby bots, mainly because the android combines the work of some of the best technology out there: Japanese humanoid robotics hardware and a Hanson Robotics head. David Hanson's android heads have received wide-spread recognition as the most human-like and expressive around.

From the news release:

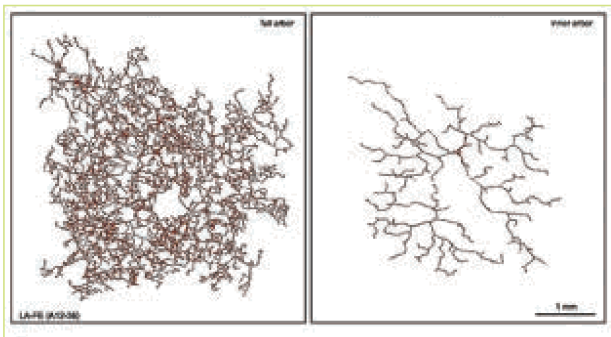
"We developed machine-learning methods to analyze face-to-face interaction between mothers and infants, to extract the underlying social controller used by infants, and to port it to Diego-San. We then analyzed the resulting interaction between Diego-San and adults. With high definition cameras in the eyes, Diego-San sees people, gestures, expressions, and uses AI modeled on human babies to learn from people the way that a baby hypothetically would. The facial expressions are important to establish a relationship, and communicate intuitively to people."

Diego-San was developed at the Machine Perception Lab and funded by the National Science Foundation. The robot is another small step towards robots that are able to interact emotionally with humans. There's still a long way to go — particularly with the development of affective systems in the robot itself. Most work to date has been on expressing simulated affects and on recognizing emotions in humans.

GETTING UNDER OUR SKIN

Skin is the human body's largest sensory organ. Understanding how it works will help roboticists create more useful android skins. Thanks to a new report announced by Johns Hopkins researchers, we're a step closer to understanding the skin's sensory system.

The scientists created detailed maps of the branching patterns of sensory nerves in mouse skin. The resulting maps revealed 10 distinct groups that seem to correspond to differences in nerve functions. For example, some nerve types gather information from a single hair follicle while others branch into groups that collect averaged information from 200 or more different locations.



The images now in hand will help scientists make more sense out of known responses to stimulation of the skin. For example, if a single nerve cell is responsible for monitoring a patch of skin a quarter of an inch square, multiple simultaneous points of pressure within that patch will only be perceived by the brain as a single signal. As explained by the researchers, this is why we can't read Braille using the skin on our backs. (Not that we would if we could.) The multiple bumps that make up a Braille symbol are within such a small area that the axon branches can't distinguish them. By contrast, each sensory axon on the fingertip occupies a much smaller territory and this permits our fingertips to accurately distinguish small objects.

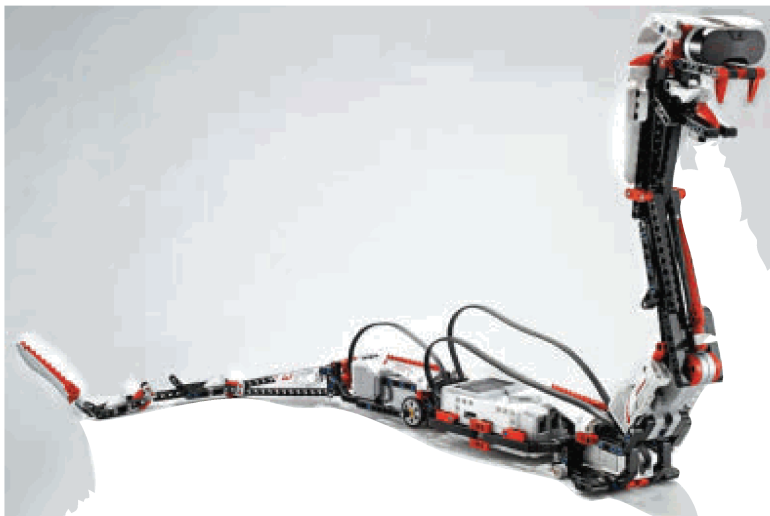
Johns Hopkins researchers also announced the discovery of strong evidence that there are specific nerve cells responsible for itch signals, which are distinct from nerves involved in pain.

AN AMAZING OPPORTUNITY

Opportunity landed at Meridiani Planum nine years ago this January. The original warranty on this robot was just a mere 90 days. That's fracking amazing!

So, what's Opportunity been up to? The robot is currently hanging out in a place called Matijevic Hill on the western rim of Endeavour Crater, some 22 miles from where Oppy touched down. Clay minerals have been detected in this area from orbit, suggesting that water may have modified the rocks, so Opportunity will be checking the place out down on the ground.

As for what's next, well, at some point this little robot is going to stop working. It's already got some quirks — nothing serious, but still, it's closing in on a decade of non-stop operation. The fact that we've got robots on Mars right now is something that should be, a continual source of wonder for every single human on Earth.



EV3 UNVEILED

The LEGO Group recently introduced LEGO® MINDSTORMS EV3 — a new platform for consumer robotics designed to introduce a younger generation to the excitement of building and programming robots while adding new flexibility for its devoted following of robotics enthusiasts.

EV3 is the result of three years of product development by an international team of LEGO designers, MINDSTORMS user enthusiasts, and leading technology experts, and will be available this summer.

EV3 comes with hundreds of LEGO bricks, plus four motors and five sensors — including a new infrared unit that can be used as robotic eyes or to allow a robot to follow a remote control.

LEGO is celebrating the 15th anniversary of the MINDSTORMS kit, and they really wanted to make the product more exciting to "an audience of children who have grown up with technology."

One of the biggest upgrades is the EV3 programmable brick — the heart of every MINDSTORMS project. The brick has been revamped internally and externally. The central processor is an ARM 9 chip, with 64 MB of RAM and 16 MB of Flash built in for storing programs. An SD slot allows the memory to be expanded, and there's a better display.

The biggest innovation is that LEGO has made it easier for the EV3 brick to communicate via Bluetooth with Android and iOS apps. That means you can use a smartphone or tablet to control a MINDSTORMS robot or give it new behaviors.

While the previous version of MINDSTORMS did offer Bluetooth connectivity, Apple devices wouldn't play with the system. Now — due to the addition of a secure Bluetooth chip — these robots can connect to iOS devices for the first time, allowing robots to be controlled from an iPhone or iPad.

There's also a USB port that allows users to connect any Wi-Fi dongle. The whole thing runs under a version of the Linux operating system, and LEGO promises to open up the system as much as possible to hackers with the release of detailed documentation and SDKs coming later.

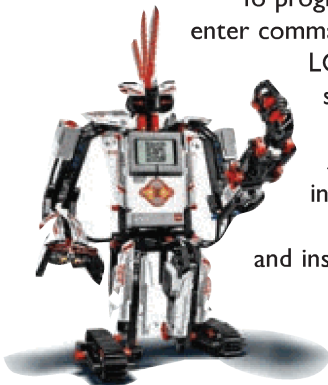
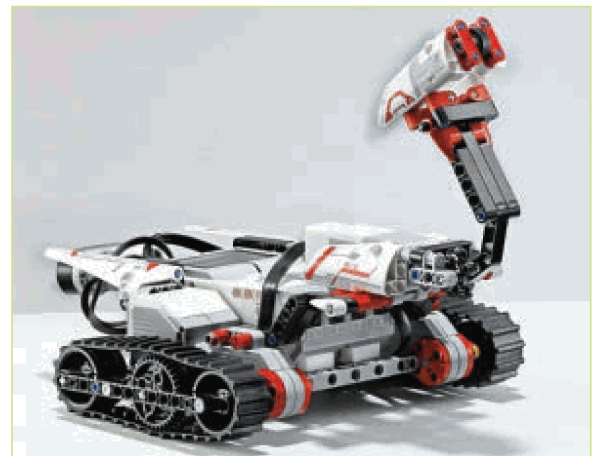
To program your robotic creations, you can enter commands directly into the EV3 brick via its LCD or you can use the easy-to-use PC software provided. The set also includes a program that LEGO created with Autodesk that shows step-by-step 3D instructions for various projects.

The EV3 set will include instructions for 17 different robots, including walking humanoids and insect-like creatures. It will sell for \$350 and will be available in the second half of this year.

In addition to the retail version, LEGO is launching an educational version, as well.

The educational version is actually a collection of kits designed for classroom use, and it features some unique components compared to the retail version, such as a gyroscope sensor that allows the construction of self-balancing robots, Segway-style.

The educational version also comes with rechargeable lithium-ion batteries for the programmable brick. It's assumed that students will work in pairs in the classroom, so this version starts at \$5,000 with enough pieces to allow 24 students per class.



COMBAT ZONE

Discuss this section in the
SERVO Magazine forums at
<http://forum.servomagazine.com>

Featured This Month:

26 *BUILD REPORT:*
Mini Bot Hockey Bots —
Part 2
by **Pete Smith**

30 *PARTS IS PARTS:*
Product Review:
OrangeRx T-SIX
2.4 GHz Transmitter
by **Pete Smith**

32 *Building a Large Scrolling*
Display — Part 2
by **Pete Smith**

31 *CARTOON*

[www.servomagazine.com/index.php
?/magazine/article/march2013
_CombatZone](http://www.servomagazine.com/index.php?/magazine/article/march2013_CombatZone)

BUILD REPORT:

Mini Bot Hockey Bots — Part 2

● by Pete Smith

In Part 1 of this series, I covered the design of a new mini bot. This time, I will show how I made the various parts that make up the chassis and drivetrain.

I had ordered 10 sets of the top and bottom panels, plus a set

of templates of the other parts to be water cut by my usual supplier, Team Whyachi www.teamwhyachi.com.

The panels are 1/16" 6061 aluminum and the templates are 1/4" steel plate (**Figure 1**).

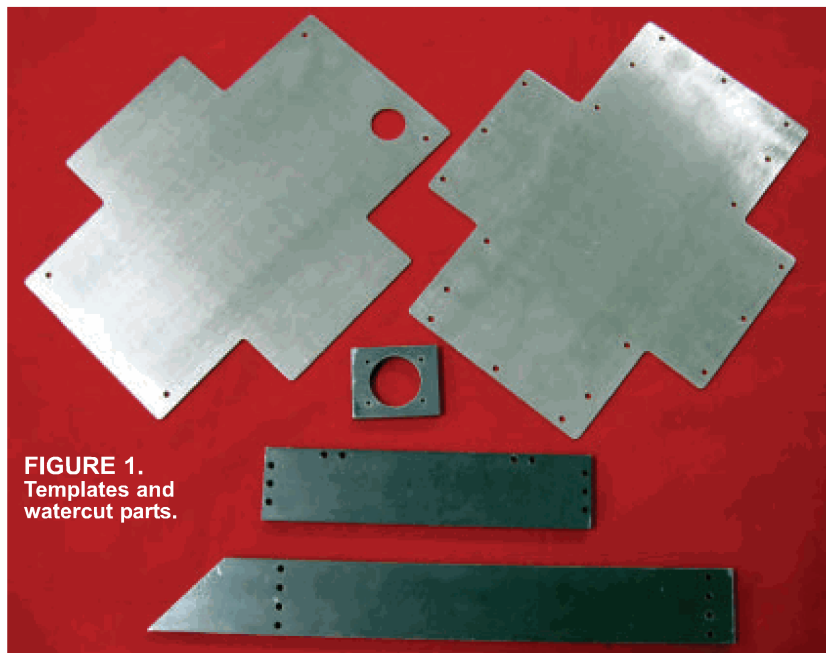


FIGURE 1.
Templates and
watercut parts.

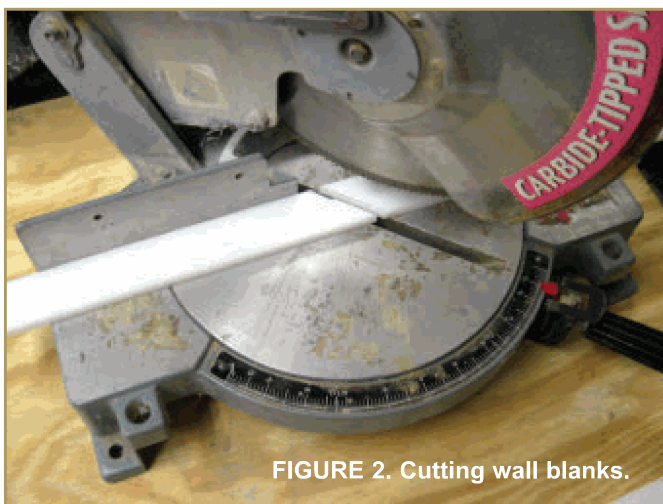


FIGURE 2. Cutting wall blanks.



FIGURE 4.
Templates
for hole
locations.

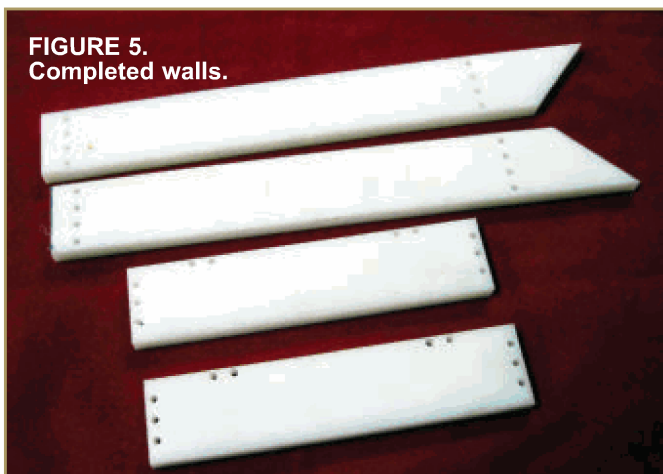


FIGURE 5.
Completed walls.



FIGURE 3.
Wall blanks.



FIGURE 6.
Forstner bit
and motor
mounts.

The design used an off-the-shelf width of UHMW (from www.mcmaster.com), so to cut the basic outline of the side and end walls I simply marked everything out using the templates, and then cut everything to size using a miter saw with a fine tooth blade (**Figure 2**). I ordered an equal amount of both black and white UHMW, so the bots would be different colors for easier team identification.

This quickly resulted in a big pile of wall blanks (**Figure 3**). The UHMW was slightly oversize, but not enough to affect the parts. The templates (**Figure 4**) were then used to add all the various mounting holes required in the part.

I then simply clamped the blank and the template

together, and drilled through all the holes in the template to produce a set of parts (**Figure 5**).

In order to make the motor mounting blocks, I used a stop fitted to the vise on my mill to consistently position the UHMW strip under a 1" forstner bit (**Figure 6**). These bits are designed for cutting wood, but they also work well on UHMW. I would position the strip, bore the hole, and then use the template to mark off the required length and cut off a mounting block (**Figure 7**). This procedure was repeated until I had produced all the required blocks.

Each block requires two holes in one side to mount it to the chassis. To accurately position these holes, I made a small jig out of a scrap block of aluminum



FIGURE 7.
Cut motor mounts
to length.

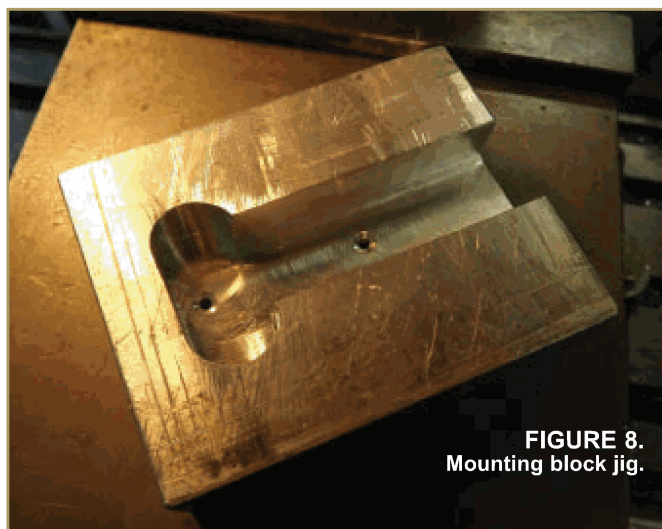


FIGURE 8.
Mounting block jig.

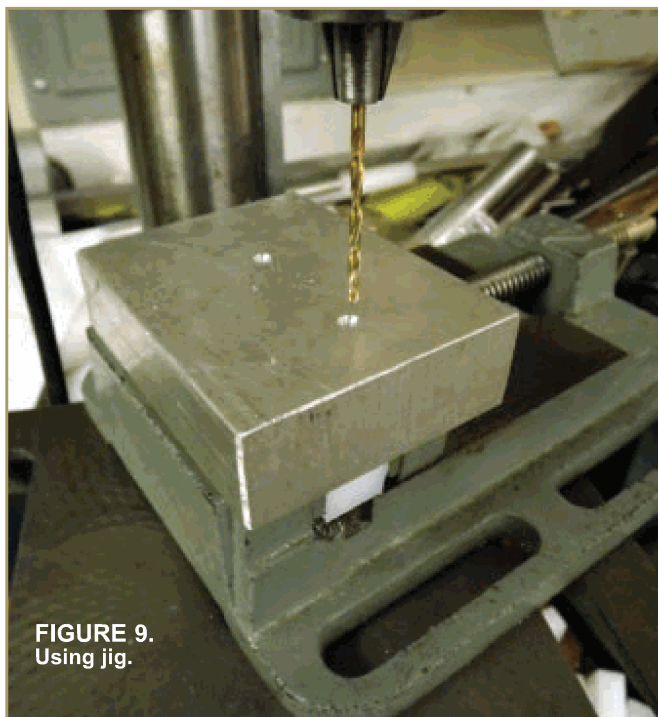


FIGURE 9.
Using jig.

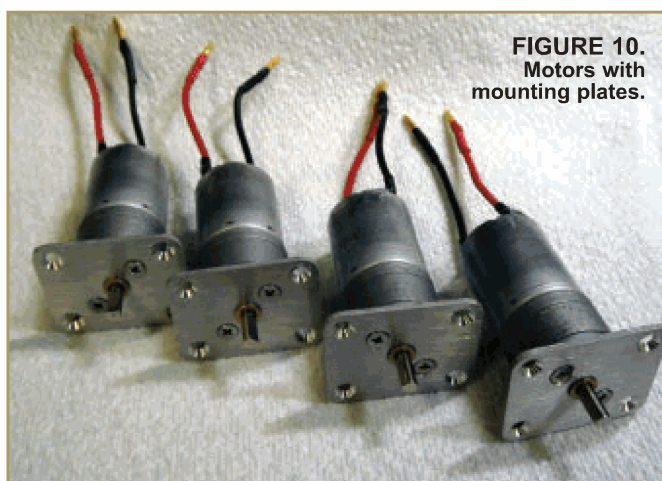


FIGURE 10.
Motors with
mounting plates.

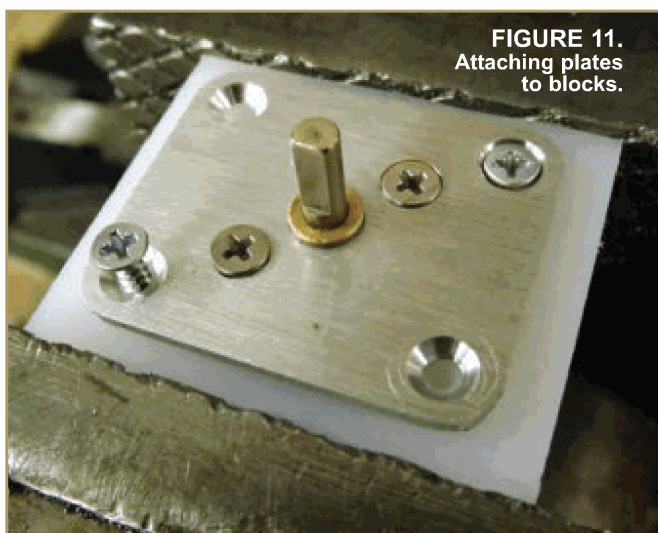


FIGURE 11.
Attaching plates
to blocks.

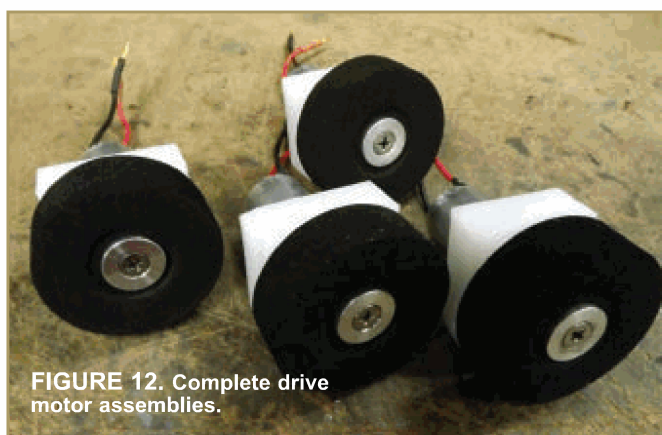


FIGURE 12. Complete drive
motor assemblies.

(Figure 8) and then drilled the holes on my drill press (Figure 9). The black blocks turned out to be just slightly wider than the white ones, so I just completed all the white blocks for now. I'll modify the jig or make a new one for the black blocks later.

I put together four Kitbots 1,000 rpm motors and mounting plates (Figure 10), then screwed them into the blocks (Figure 11). The holes in the plates themselves

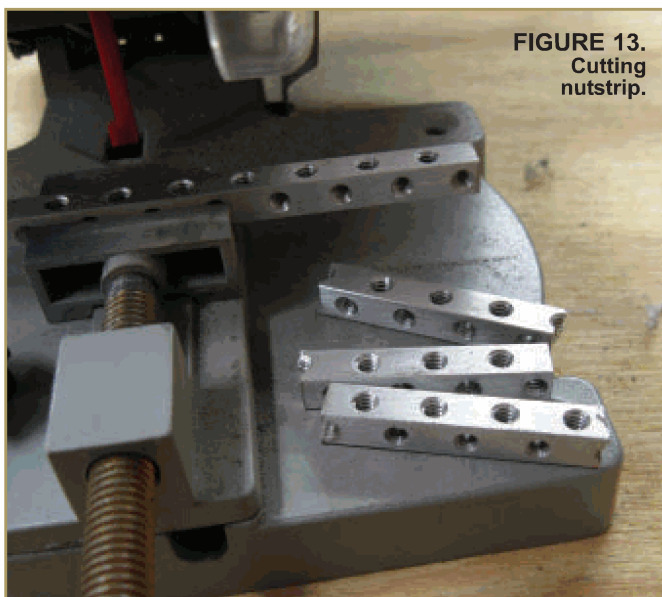


FIGURE 13.
Cutting
nutstrip.

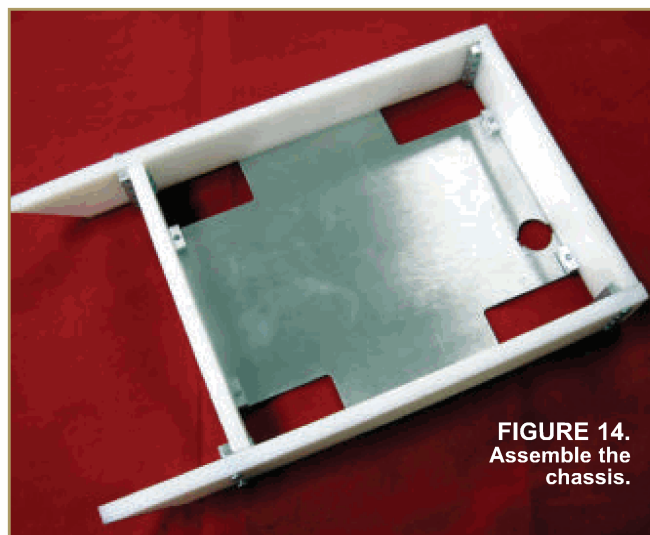


FIGURE 14.
Assemble the
chassis.

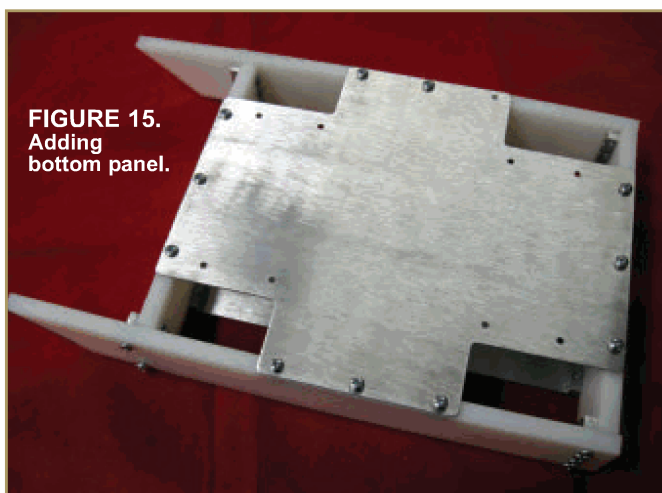


FIGURE 15.
Adding
bottom panel.

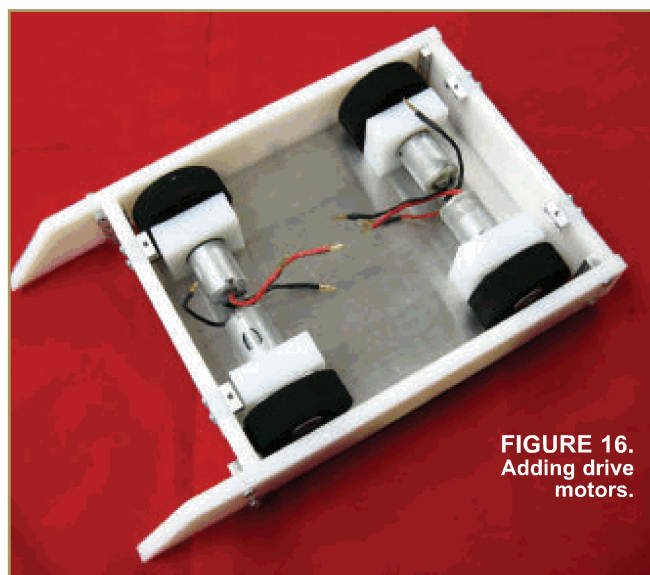


FIGURE 16.
Adding drive
motors.

act as a template for the drilled pilot holes. Just remember to drill and fit one screw before drilling the other three. That helps to make sure they are all correctly aligned. I added Dave's Hubs and 2.25" Lite Flite wheels to complete the motor assemblies (**Figure 12**).

The chassis uses Kitbots mini nutstrip to hold it together, and I rough-cut the sections on my small miter saw (**Figure 13**), then filed the ends smooth.

The chassis is then assembled using 6-32 screws. With the top panel fitted to keep it square (**Figure 14**), the bottom panel can then be added. It is secured by placing 12 #6 self-tapping screws into the drilled pilot holes (**Figure 15**). Line up the panel and insert a couple of the screws before drilling and inserting the rest.

More #6 self-tapping screws are used to secure the four motor assemblies (**Figure 16**); refitting the top panel (**Figure 17**) completes the rolling chassis.

The final part of this series will cover the installation

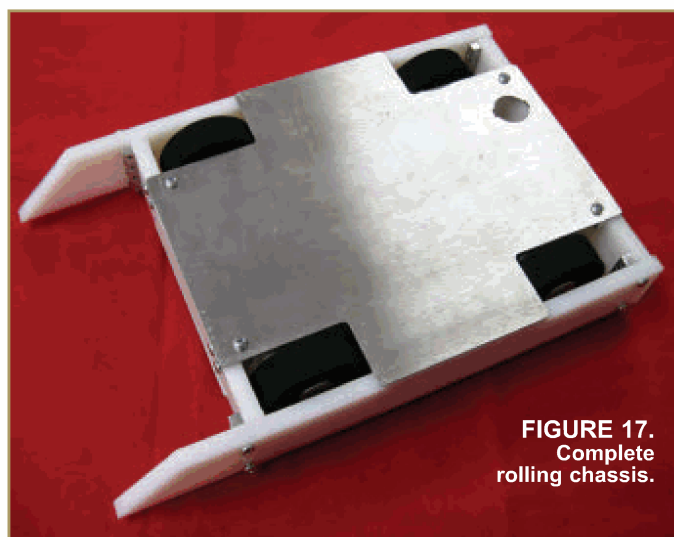


FIGURE 17.
Complete
rolling chassis.

of the electronics and setting up the radio transmitter and receiver. **SV**

PARTS IS PARTS:

Product Review: OrangeRx T-SIX 2.4 GHz Transmitter

● by Pete Smith

The OrangeRx range of 2.4 GHz spread spectrum DSM2 receivers (available from HobbyKing) has proved to be very useful in combat robotics over the last couple of years as they provide a cheaper alternative to the Spektrum receivers required by the popular DX6e and DX6i transmitters.

They are also available in a wider range of sizes (as you can see in **Figure 1**), so can easily fit bots in all weight classes.

To complement these receivers, OrangeRx has now released a fully programmable six-channel DSM2 transmitter. It is priced at about \$85 (delivered to the US), so is competitive with the non-programmable DX5e (\$70), and is half the cost of the programmable DX6i (\$170).

The OrangeRx T-SIX (**Figure 2**) comes well packaged and includes a manual, but no batteries or receivers. It uses four AA batteries (alkaline or rechargeable) or a 2S LiPo pack. I used alkalines for this review.

The radio is easy to bind to receivers, program, and use. The only real issue I found is with the manual, which is not very well written. Fortunately, this is not much of a problem since the radio itself has simple clear menus, plus a big clear backlit LCD display with easy to use buttons and a roller for scrolling. I very quickly worked out how to do all the basic functions required to program it for a simple bot.

It failsafes correctly for combat robots, and the 10 model memory should be enough for most users.

The OrangeRx T-SIX brings fully programmable transmitters into the reach of most builders and its ease of use is a great bonus. **SV**

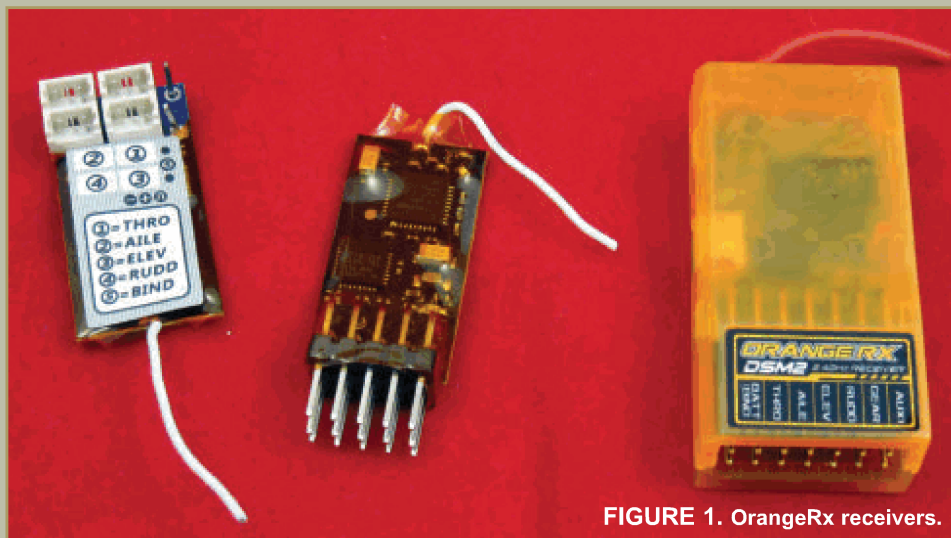


FIGURE 1. OrangeRx receivers.



FIGURE 2. OrangeRx transmitter.

Sneak Preview

Coming in April's *Combat Zone*: **SERVO** unveils an entirely new class of fighting robots!

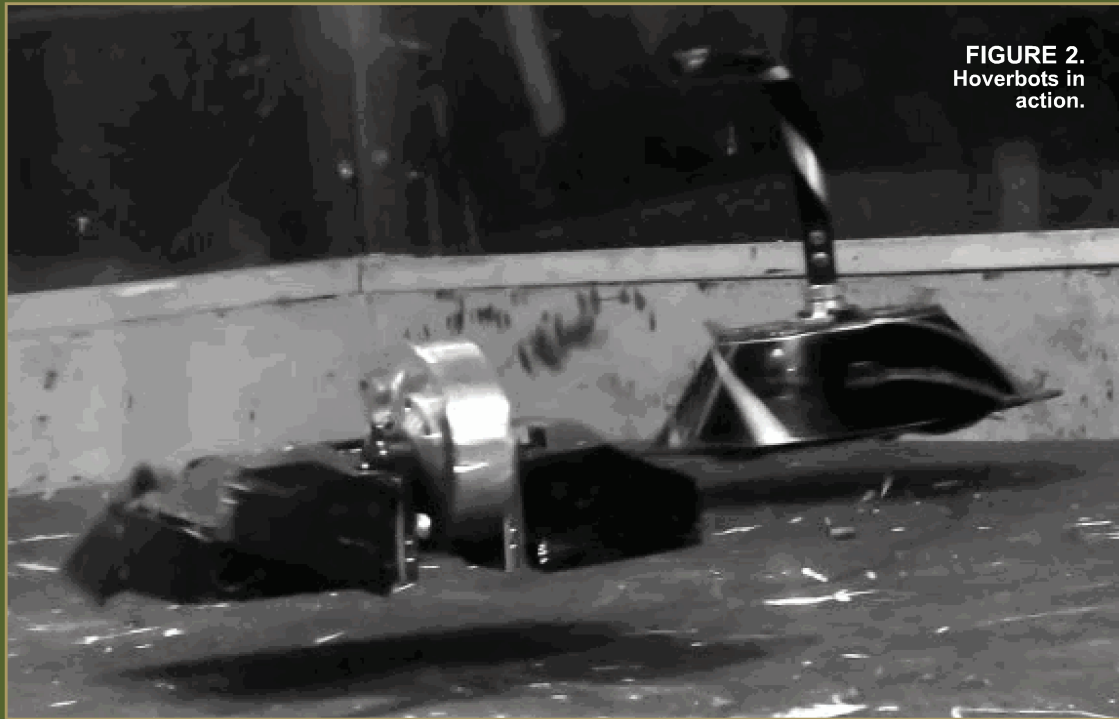


FIGURE 2.
Hoverbots in
action.

Melty Brains

by Kevin Berry

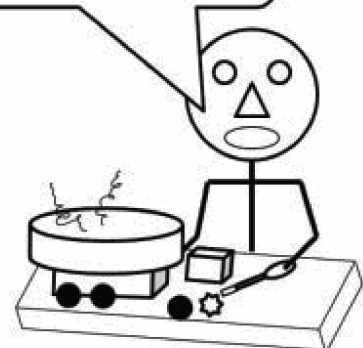
hom·o·nym ('hämə,nim) (n.): Two or more words having the same pronunciation but different meanings. Ex: "March Madness"

Most Sports Nuts:



Servo Readers:

OMG! Only 3 weeks until
RoboGames!



Building a Large Scrolling Display –

Part 2: Adding a Second Panel

● by Pete Smith

Buoyed up by my success in getting one panel to work, I decided to add a second panel to increase the impact of the display and to make it a little

easier to read.

You can run two panels off the Freetronics Eleven simply by using the provided signal cable to link the second panel to the first as shown

in the middle of **Figure 1** (ignore the other wiring for now), and making a small change to the code so that the microcontroller recognizes the second panel.

You simply change the command “#define DISPLAYS_ACROSS 1” to “#define DISPLAYS_ACROSS 2.” Load the code onto the Freetronics Eleven as before; now the display will run across both panels, but at a reduced brightness.

To restore the panels to full brightness, you need to add a 5V power source directly to them. I bought a small 5V 3A power supply on eBay for about \$10 (**Figure 2**), then connected a power cord to it and output wires that feed both panels (refer again to **Figure 1**).

CAUTION: Line voltages are dangerous and can injure or kill.

If you are not experienced with how to safely connect and shield line voltages, then get a laptop or wall wart type power supply. Mine will be inside a protective box in the final display so that the high voltages will not be accessible.

The display, however, currently still needs the computer to be connected to power the microcontroller. To be able to run the display without the computer, you need to be able to replace the 5V provided by the USB cable.

At first glance, the Eleven

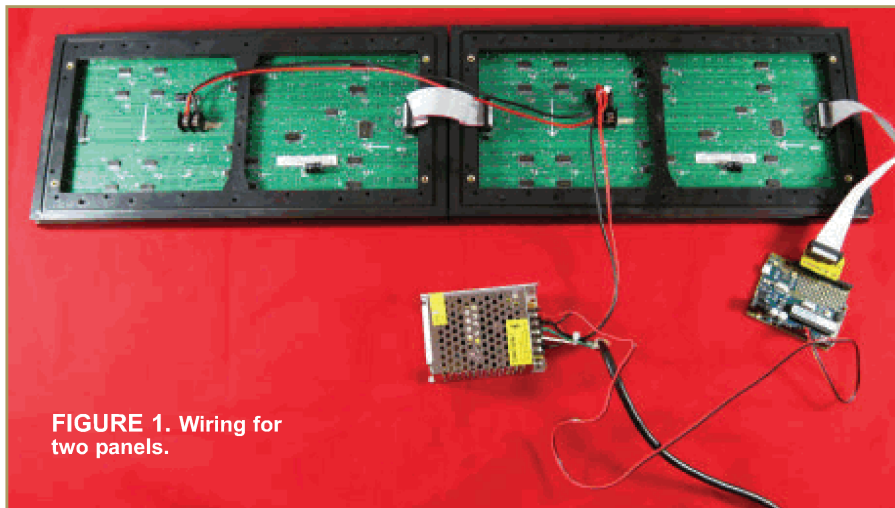


FIGURE 1. Wiring for two panels.

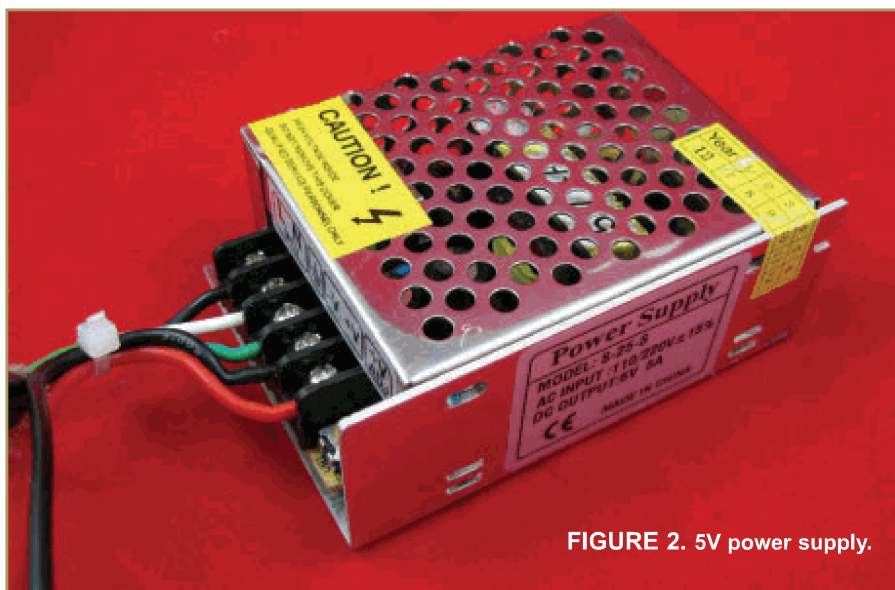


FIGURE 2. 5V power supply.

needs a higher voltage to run. However, after asking about this on the Freetronics forum, I was told that I could simply supply 5V from the DMD's power supply to the 5V and GND pins of the power connector on the board.

I modified an old signal extension cable so that I could simply connect one end to the 5V power supply and plug the other into the microcontroller as in **Figure 3**. Make sure you get the 5V and ground the right way around ... I'm not sure what happens if you don't, but I doubt it would be good!

Unplug the USB cable and then switch on the 5V supply. The display will run the last program loaded independently of the PC that loaded it.

To load a new program, simply switch off the 5V power and plug in the USB cable as before.

The full display can be seen running in **Figure 4**. It's bright enough for any indoor application, and is still readable even in

sunlight outside.

I plan to build a case to protect the panels, the power supply, and the microcontroller so

that it's easy and safe to mount on a wall or a combat robot arena. It will get its first use at the Motorama event in February. **SV**

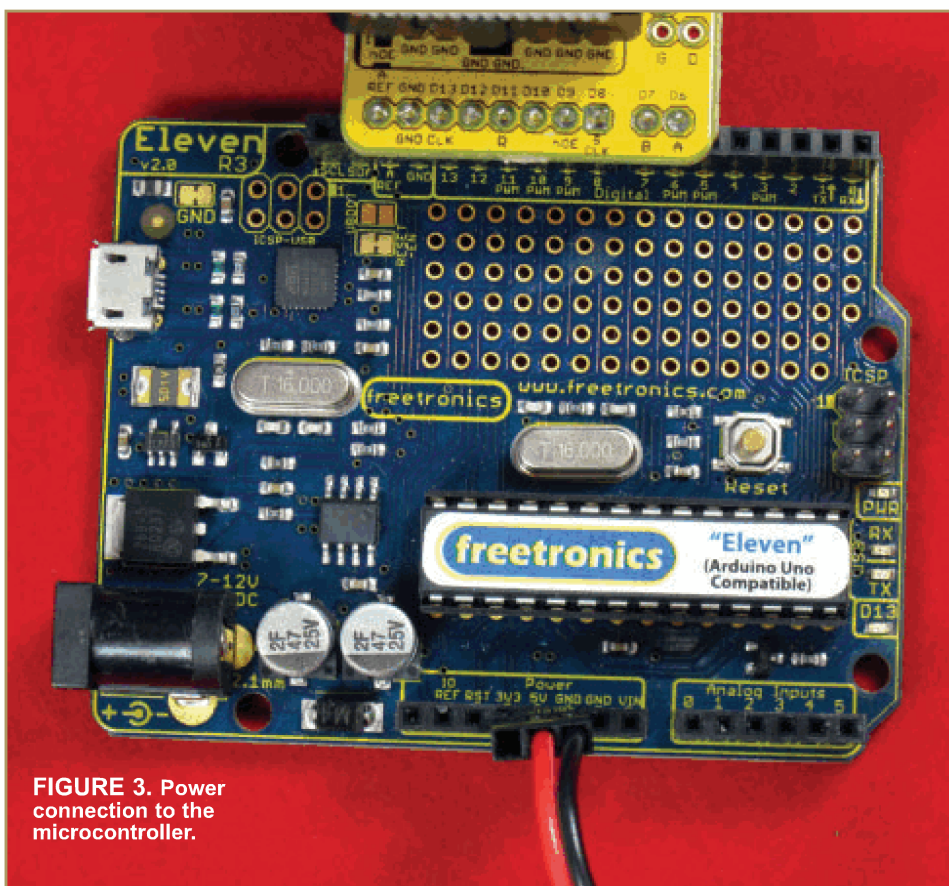


FIGURE 3. Power connection to the microcontroller.

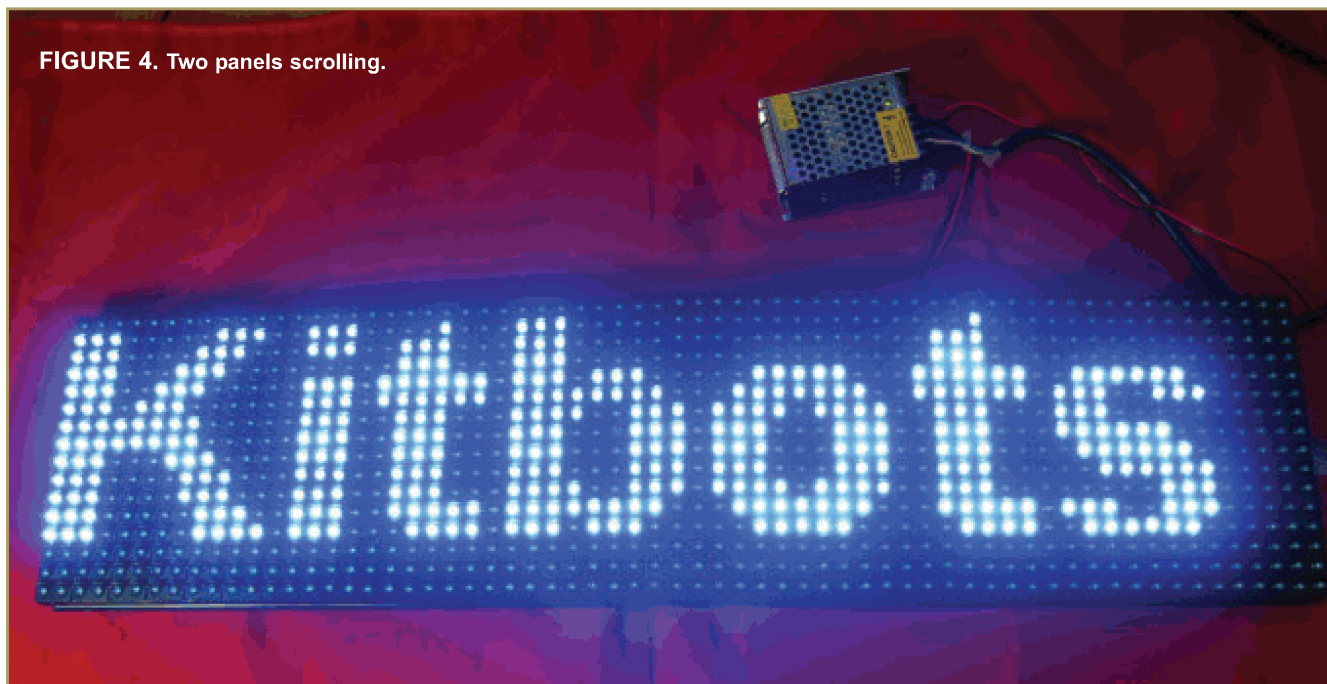


FIGURE 4. Two panels scrolling.

ROBOGAMES



Compete at RoboGames!

Last year, over 1000 builders from around the world brought over 800 robots to San Francisco, in the 4th annual international event. This year, we expect even more robots and engineers to compete. Be one! With 80 different events, there's a competition for everyone - combat, androids, sumo, soccer, Lego, art, micromouse, BEAM, or Tetsujin! More than half the events are autonomous. Even if you just come to watch, you'll be overwhelmed with the diversity.

Last year, RoboGames hosted teams with over 800 robots from Argentina, Australia, Austria, Brazil, Canada, China, Colombia, Czech Republic, Denmark, Germany, India, Indonesia, Iran, Japan, Korea, Mexico, Netherlands, Peru, Singapore, Slovenia, Sweden, Taiwan, UK, and the USA.

Be a RoboGames Sponsor!

RoboGames is the world's largest open robot competition - letting people of any age, gender, nationality, or affiliation compete. Sponsoring RoboGames not only helps more people to compete, but also gets your company unrivaled press coverage and visibility. The event has been covered by CNN, ESPN, Fox, CBS, ABC, NBC (live), EBS Korea, NHK Japan, BBC, and countless print and web companies. Your logo can be everywhere the cameras turn!

Rent a Booth!

Booth spaces are at the front of the venue, ensuring lots of traffic. With 3000-5000 people each day, your company will get amazing traffic!

World's Largest Robot Competition
-*Guinness Book of Records*

North America's Top Ten Geek Fests
-*Wired Magazine*

SportCenter's Top Ten
-*ESPN SportsCenter*

"If you are a robot enthusiast, I would definitely encourage you to attend the RoboGames... Take a plane, train, space elevator, but definitely go!"

-*Servo Magazine*

"Impossible to Imagine, Impossible to Forget!"

-*Robot Magazine*

Events:

Combat: 340 lbs, 220, 120, 60, 30, 3, & 1 lbs

Androids: Wrestling, Demonstration, Stair Climbing, The Eagle, Door Opening, The Toss, Basketball, Lift and Carry, Marathon, Obstacle Run, Penalty Kick, Dash, 3:3 Soccer, Weight Lifting

Open Events: Fire-Fighting, Robomagellan, Maze/MicroMouse, Walker Challenge, Biped Race, Robot Triathlon, Line Slalom, Ribbon Climber, Vex Open, Lego Challenge, Lego Open, Aibo Performer, Balancer Race, Best of Show, Bot Hockey

Sumo: 3kg - Auto & R/C, 500g, 100g, 25g, Humanoid

Robot Soccer: Biped 3:3 & 5:5, Mirobot 5:5 & 11:11

Junior League: Lego Challenge, Lego Open, Lego Magellan, Woots & Snarks, Handy Board Ball, BotsketBall, 500 g Sumo, 120 lb combat, Best of Show, Vex Open

Tetsujin (ExoSkeleton): Lifting, Walking, Carrying

Art Bots: Static, Kinetic, Bartending, Musical, Drawing

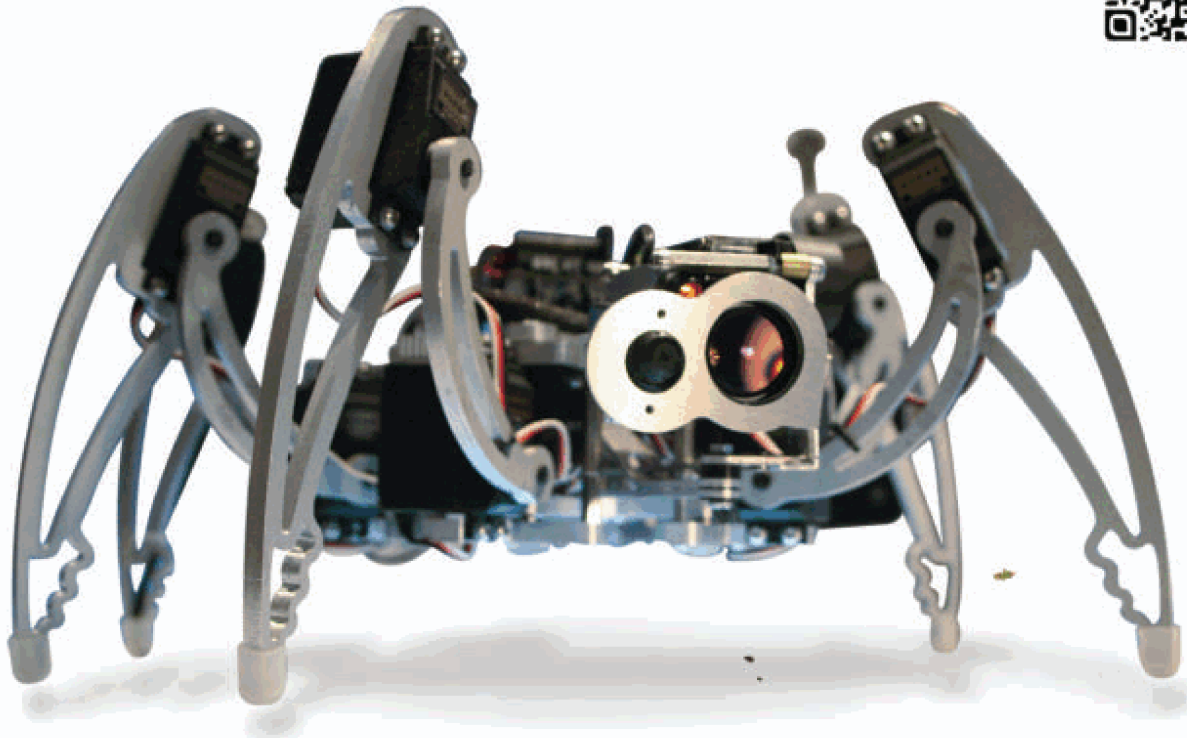
BEAM: Speeder, Photovore, RoboSapien Hacker



April 19-21, 2013

San Mateo, CA

<http://RoboGames.Net>



GENUS: SPIDERUS ROBOTICS

DESIGN MACHINES THAT SPIN THEIR OWN REALITY.

The evolving field of robotics is growing fast. It requires creative and innovative thinkers with a sound education in cutting edge advancing technology.

Start a life of innovation with a degree in Robotics and Embedded Systems from UAT.



uat.edu/robotics

Advancing Computer Science > **Artificial Life Programming** > Digital Media > Digital Video > Enterprise Software Development > Game Art and Animation > Game Design
Game Programming > Human-Computer Interaction > Network Engineering > Network Security > **Open Source Technologies** > **Robotics and Embedded Systems** > Serious Game and Simulation
Strategic Technology Development > Technology Forensics > Technology Product Design > Technology Studies > Virtual Modeling and Design > Web and Social Media Technologies

Please see www.uat.edu/fastfacts for the latest information about degree program performance, placement and costs.

Talking With Light

by Gordon McComb

Whenever you program your robot, you're telling it what to do. So, how about when your robot needs to talk back? You might use sound to allow your robot to communicate with you, like I covered back in the March through June 2012 issues of *SERVO*. Or, you can use a more direct — and often easier — way: with good old fashioned light.

Using light to provide feedback for your robot can be as simple as a single LED, or more complex using multiple LED colors. A benefit of talking with light is that it also attracts attention. It can bridge the

psychological gap between machine and person. Your robot can draw in humans by using a variety of bright lights and dazzling colors.

In this article, we'll discuss simple techniques to communicate using light with your Arduino-based robot. The projects are simple to do, with only basic components and wiring needed. Most cost under a couple dollars.

Communicating With LEDs

One light emitting diode is all it takes for your robot to communicate with you. The language may not be elegant and the conversations are amazingly short, but it gets the job done. When you don't need a talkative bot you can use a single LED, or for more words, you can use multiple LEDs or seven-segment LED display panels.

The basic LED feedback circuit is shown in **Figure 1**.

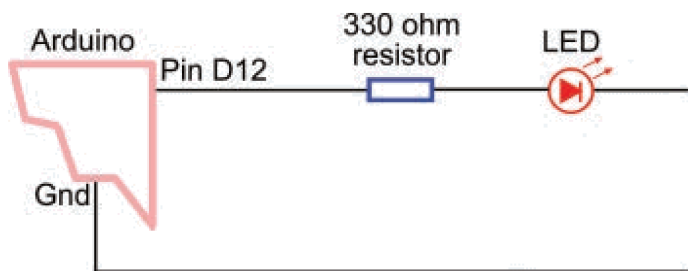


FIGURE 1. Basic connection for illuminating an LED via a microcontroller pin. Bringing the pin HIGH turns the LED on.

Listing 1.

```
void setup() {  
  pinMode(12, OUTPUT);      // Make D12 an output  
}  
  
void loop() {  
  digitalWrite(12, HIGH);   // Turn LED on  
  delay(500);               // Wait 1/2 second (500 ms)  
  digitalWrite(12, LOW);    // Turn LED off  
  delay(500);  
}
```

Sequence	Meaning
Short-Short-Short	Status A-OK
Long-Long-Long	Unknown trouble
Long-Short-Long	Battery low
Short-Long-Short	Cannot read sensors
Short-Long-Long	Goal reached
... and so forth	

Table 1.

It uses just one of the I/O pins of your robot's Arduino, along with a current-limiting resistor and an LED. Code running in your Arduino is demonstrated in **Listing 1**. It merely turns the LED off. Vary the *delay* value to make the LED flash on and off at a faster or slower rate. This sketch is just a variation of the Blink example that comes with the Arduino software.

Remember that you're not limited to merely lighting an LED to show status. You can flash the LED using different patterns to communicate more variations. Use Morse Code to relay a variety of conditions or to "speak" phrases. If you don't know Morse Code, you can invent your own system. **Table 1** shows just a few ideas. Note that in each case, there's an *Off* pause between the three-flash sequence.

Put the LED where it's easy to see, and select a component large and bright enough to make it visible from across the room — the LED connected to pin 13 on the Arduino just doesn't cut it in most instances. I like to use large 5 mm bright red or yellow LEDs mounted on the top of the robot that can be seen at any angle.

Use multiple LEDs when you want to quickly convey operating or sensor status. For example, you might light an LED each time one of the bump switches or proximity detectors sense an object. You can then see how the robot responds to the condition, such as running your "back up and turn the other way" code. You know there's a problem if an LED doesn't light when it should, or lights up but the robot doesn't take the proper action.

When only a small number of LEDs are needed, you can wire them directly to the I/O pins of your microcontroller. You simply duplicate the circuit and code from above, and use a different output pin assignment for each LED.

If you want to use more than four or five LEDs, then you probably don't want to dedicate an I/O pin for each one. With a simple serial in parallel out (SIPO) shift register, you can turn three pins into eight. Refer to **Figure 2** for a schematic using a 74595 SIPO integrated circuit. This chip is widely available and inexpensive at under \$1; you can select any member of the '595 family such as the 74HC595 or 74HCT595 — whatever is available to you.

Sample program code for the Arduino is shown in **Listing 2**. The shift register works by first setting the *latch* line to LOW, and keeping it there for the time being. Then, eight bits of data is sent bit by bit to the *data* pin of the

Listing 2.

```
int latchPin = 8;      // Connected to Latch pin
int clockPin = 12;     // Connected to Clock pin
int dataPin = 11;      // Connected to Data pin

void setup() {
  // Set all pins to output
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
}

void loop() {
  // Count 0 to 255
  for (int val = 0; val <= 255; val++) {

    // Disable update during count
    digitalWrite(latchPin, LOW);
    // Shift out bits
    shiftOut(dataPin, clockPin, MSBFIRST, val);

    // Activate LEDs
    digitalWrite(latchPin, HIGH);

    // Short delay to next update
    delay(100);
  }
}
```

Pin A	Pin B	LED Output*
LOW	LOW	Off
LOW	HIGH	Green
HIGH	LOW	Red
Pulse**	Pulse**	Orange

* The actual color depends on how you connect the LED. Some bi-color LEDs display other than red and green.

** When pulsing, one pin is LOW while the other is HIGH. Pulse at a rate of at least 10 times per second. That way, your eyes will blend the flashes into a single combination color.

Table 2

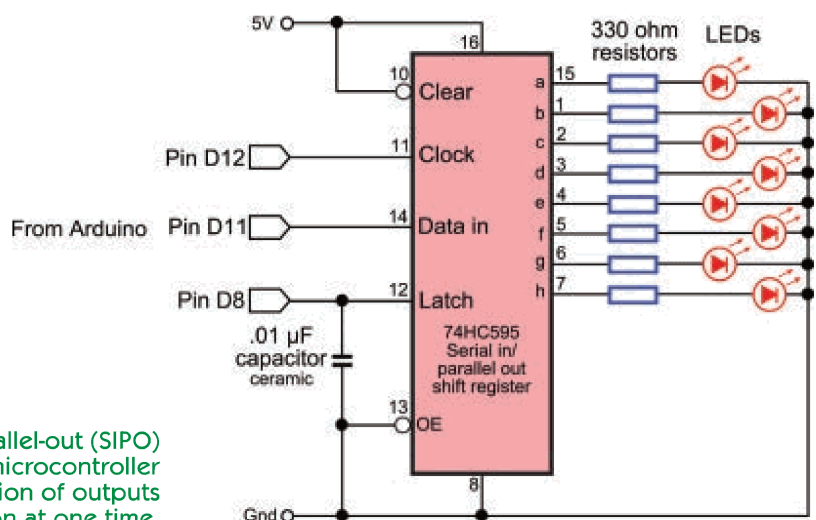


FIGURE 2. A 74595 serial-in, parallel-out (SIPO) shift register converts just a few microcontroller I/O lines to eight outputs. Any combination of outputs can be turned on at one time.

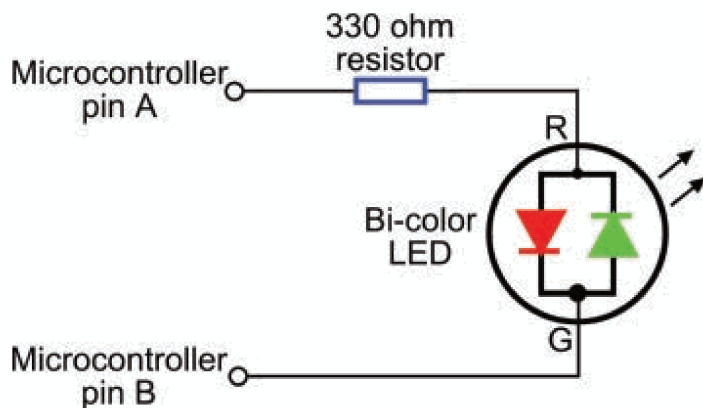


FIGURE 3. Connecting a bi-color LED to two I/O pins of a microcontroller. To display one of the colors, make its pin HIGH. Quickly alternating between the two pins produces a combination hue of the two colors combined.

Pin A	Pin B	LED Output*
LOW	LOW	Off
LOW	HIGH	Green
HIGH	LOW	Red
HIGH	HIGH	Orange

* Actual colors depend on the LED you're using.

Table 3.

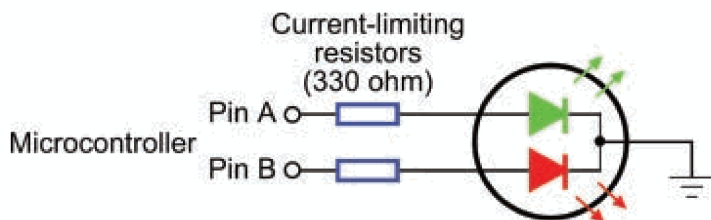
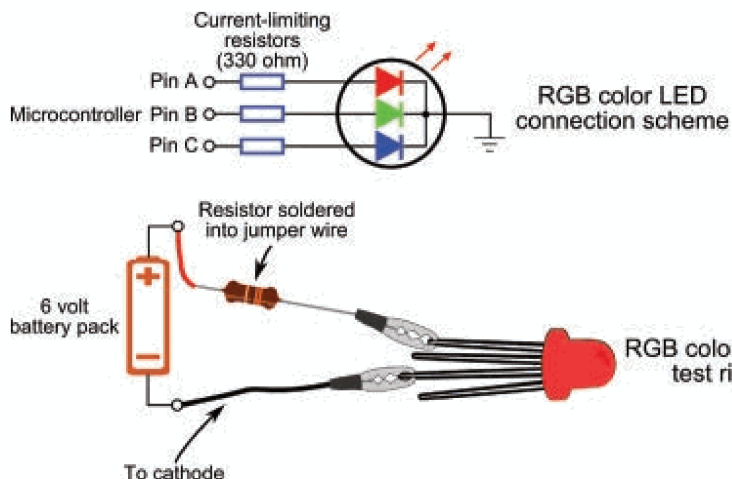


FIGURE 4. Connecting a tri-color LED to two I/O pins of a microcontroller. Either or both of the diodes can be turned on at once.



Listing 3.

```
int redPin = 9;      // Red diode of LED
int greenPin = 10;   // Green diode of LED

void setup() {
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
}

void loop() {
  digitalWrite(redPin, HIGH);
  delay(500);
  digitalWrite(redPin, LOW);
  delay(500);

  digitalWrite(greenPin, HIGH);
  delay(500);
  digitalWrite(greenPin, LOW);
  delay(500);

  digitalWrite(redPin, HIGH);
  digitalWrite(greenPin, HIGH);
  delay(500);
  digitalWrite(redPin, LOW);
  digitalWrite(greenPin, LOW);
  delay(500);
}
```

chip. For each bit, the *clock* pin on the '595 is toggled to tell the chip that new data has been sent.

The Arduino makes sending serial data easy because it packages up the data and clock activities in one simple statement: *shiftOut*. To use this statement, you specify the number of the *data* and *clock* pin, the order of the data to be sent, and the value — from 0 to 255 — you wish to use.

```
shiftOut(dataPin, clockPin,
MSBFIRST, numberValue);
```

After all the shifted out data has been sent, the program returns the *latch* pin to HIGH. This sets the output pins of the 74595 chip, illuminating the LEDs as desired.

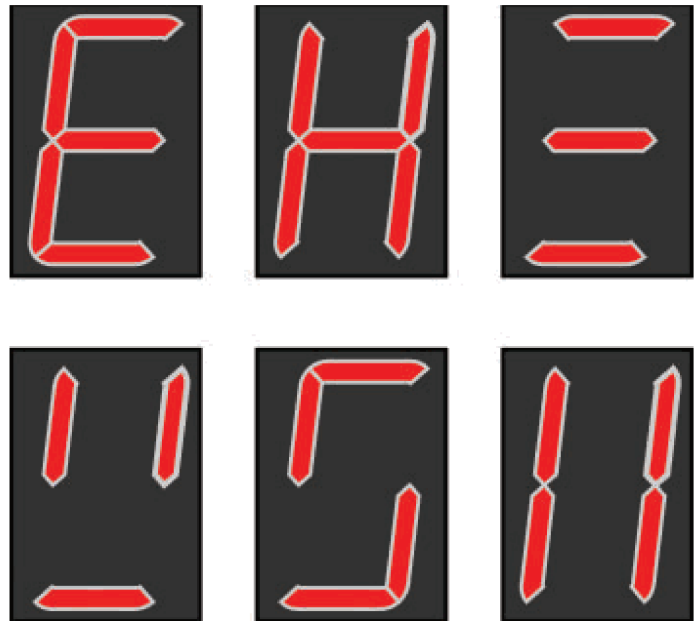
Robo Talk With Multi-color LEDs

Some LEDs are engineered to produce more than one color.

- *Bi-color* LEDs contain red and green LED elements. You control which color is shown by

FIGURE 5. Connecting a multi-color (red-green-blue) LED to three I/O pins. Any color in the rainbow can be reproduced by altering the brightness of the three diodes.

FIGURE 6. Unusual non-numeric shapes can be created by activating selected segments of a seven-segment LED display. Use this system to display numbers (of course) or codes.



reversing the voltage to the LED. You can also produce a yellowish-orange by quickly alternating the voltage polarity.

- *Tri-color* LEDs are functionally identical to bi-color LEDs, except that they have separate connections for the red and green diode.

- *Multi-color* LEDs contain red, green, and blue LED elements. You control which color to show by individually applying current to separate terminals on the LED.

Figure 3 shows how to connect a bi-color LED to two Arduino pins. To turn the LED off, put both pins LOW. To turn on one color or the other, put one of the pins HIGH. Refer to **Table 2**.

As an example, the following code for the Arduino toggles between the two colors of a bi-color LED:

```
digitalWrite(led1, HIGH);
digitalWrite(led2, LOW);
delay(1000);
digitalWrite(led1, LOW);
digitalWrite(led2, HIGH);
delay(1000);
```

For the mixed color effect, simply toggle between the two colors more quickly. Try reducing the delay from 1,000 milliseconds (one second) to 10 milliseconds (1/100th of a second).

Only one diode in a bi-color LED can be on at any time. Recall that in a tri-color LED, each diode has its own connection lead so you can switch either one on or off. You can also have both on at the same time, mixing the colors together. **Figure 4** shows how to connect a tri-color (three lead) LED to the Arduino (or other microcontroller). Note that each diode in the LED gets its own current-limiting resistor.

See **Listing 3** for a short demonstration program of toggling a tri-color LED from red

to green to orange. Refer to **Table 3**.

Finally, multi- or RGB-color LEDs have a red, green, and blue diode. These three primary colors can be displayed independently, or in different combinations to produce many other colors. You use these the same as with a tri-color LED, except that you need a third I/O pin to control the additional color.

Figure 5 shows the connection scheme for a multi-color LED, plus a testing rig so you can experiment with how they work. Solder a current-limiting resistor

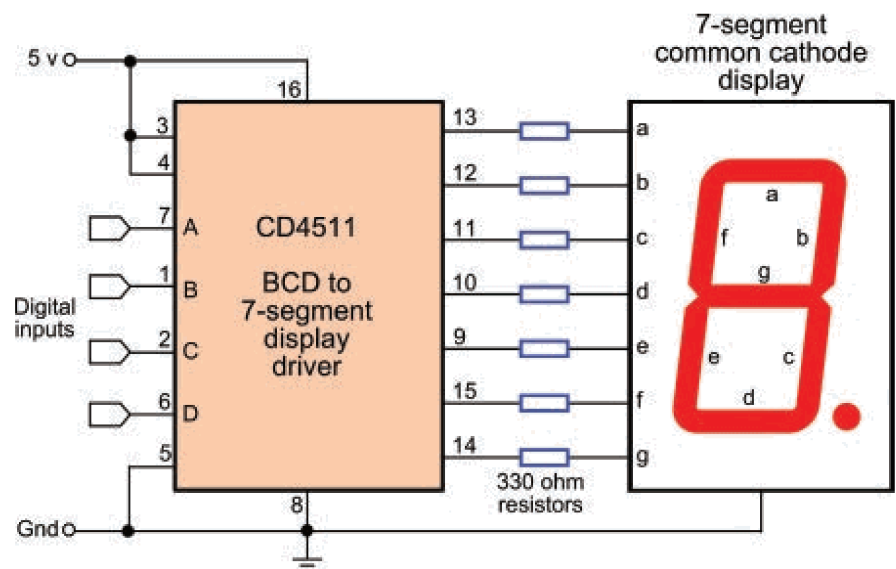


FIGURE 7. Connection diagram for a CD4511 (binary coded decimal) to seven-segment driver. Numbers appear in the display based on the binary coded values on the A to D input pins.

Listing 4.

```
int redPin = 9;           // Red diode of LED
int greenPin = 10;        // Green diode of LED
int bluePin = 11;         // Blue diode of LED

void setup() {
  // Start with everything off
  analogWrite(redPin, 0);
  analogWrite(greenPin, 0);
  analogWrite(bluePin, 0);
}

void loop() {
  // Fade red in and out
  for(int fadeValue = 0 ; fadeValue <= 255; fadeValue++) {
    analogWrite(redPin, fadeValue);
    delay(5);
  }
  for(int fadeValue = 255 ; fadeValue >= 0; fadeValue--) {
    analogWrite(redPin, fadeValue);
    delay(5);
  }

  // Fade in blue, then add green, and finally red
  for(int fadeValue = 0 ; fadeValue <= 255; fadeValue++) {
    analogWrite(bluePin, fadeValue);
    delay(5);
  }
  for(int fadeValue = 0 ; fadeValue <= 255; fadeValue++) {
    analogWrite(greenPin, fadeValue);
    delay(5);
  }
  for(int fadeValue = 0 ; fadeValue <= 255; fadeValue++) {
    analogWrite(redPin, fadeValue);
    delay(5);
  }

  // Wait 2 seconds before doing it all again
  // Note that all three colors may not create a
  // pure "white" because the diodes in the LED
  // do not overlap one another
  delay(2000);

  analogWrite(redPin, 0);
  analogWrite(greenPin, 0);
  analogWrite(bluePin, 0);
  delay(500);
}
```

(for example, 330 to 470 Ω) inline with the jumper wire. Connect the – terminal of the battery to the cathode lead of the LED, and then touch each of the three anode leads in turn to watch the colors.

Recall that you can mix colors by turning on more than one diode in the LED at a time. Thanks to the Arduino's PWM (pulse width modulation) feature, you can vary the intensity of the light, and create thousands of color combinations. The Arduino sketch in **Listing 4** shows how to use the *analogWrite* statement which produces a PWM signal on specific pins of the microcontroller. We'll be using digital pins D9, D10, and D11 for connecting to a multi-color LED (don't forget the current-limiting resistors!).

Note: The hookup diagrams for the tri- and multi-color LEDs show common cathode devices — that is, the cathode (negative) end of all the diodes in the device are tied together. Multiple color LEDs are also available in common anode style, where the anode (positive) ends are linked together. The concept of using these are the same though, of course, you must reverse the wiring. The LED is turned on when the cathode connection is brought LOW.

Listing 5.

```
int outA = 8;           // Connected to A pin on 4511
int outB = 9;           // Connected to B pin
int outC = 10;          // Connected to C pin
int outD = 11;          // Connected to D pin

void setup() {
  // Set all pins to output
  pinMode(outA, OUTPUT);
  pinMode(outB, OUTPUT);
  pinMode(outC, OUTPUT);
  pinMode(outD, OUTPUT);
}

void loop() {
  // Display 3
  digitalWrite(outA, HIGH);
  digitalWrite(outB, HIGH);
  digitalWrite(outC, LOW);
  digitalWrite(outD, LOW);
  delay (1000);

  // Display 9
  digitalWrite(outA, HIGH);
  digitalWrite(outB, LOW);
  digitalWrite(outC, LOW);
  digitalWrite(outD, HIGH);
  delay (1000);
}
```

	4511 Inputs					
BCD	A	B	C	D	4511 Outputs	Numeral
0000	0	0	0	0	1111110	0
0001	1	0	0	0	0110000	1
0010	0	1	0	0	1101101	2
0011	1	1	0	1	1111001	3
0100	0	0	1	0	0110011	4
0101	1	0	1	0	1011011	5
0110	0	1	1	0	0011111	6
0111	1	1	1	0	1110000	7
1000	0	0	0	1	1111111	8
1001	1	0	0	1	1110011	9

Table 4.

FIGURE 8. The 74595 SIPO shift register driving a seven-segment LED display. Using the shift register allows any combination of segments to light.

Linguistics With Seven-Segment LED Displays

Your robot can also talk to you using one or more seven-segment numerical LED displays. You can light up the segments to produce numerals, in which case your robot can output up to 10 “codes” to indicate its status. For instance, 0 might be OK, 1 battery low, and so on.

You can illuminate the segments to make non-numeric shapes; turn on each segment individually or in combination. **Figure 6** shows some variations, including an E for Error, H for Help, and numerous symbols that can mean things like which motor is supposed to be on, or a battery level indicator.

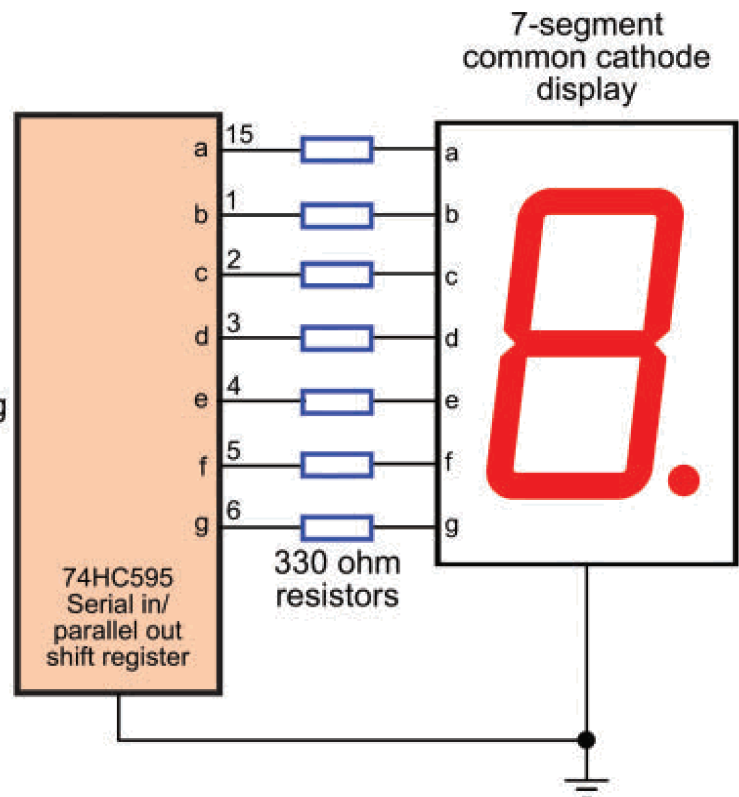
The easiest way to show numerals on a seven-segment display is to use a display driver IC, such as the CD4511 or 7447. These chips have four inputs and eight outputs — seven outputs for the numeric segments, and an eighth for the decimal point which we won’t be using in this example.

To display a number, set its binary coded decimal (BCD) value at the A-D input lines. For example, to display a 3, use the binary value 0011. Refer to **Table 4**.

Refer to **Figure 7** for a hookup diagram for the CD4511 to a common cathode seven-segment LED display. **Listing 5** is a simplistic Arduino program that sends different values to the CD4511, lighting up different segments. (You can use the same general concept with individual LEDs.) The circuit in **Figure 7** shows a common cathode seven-segment LED module; that is, all the LED segments in the display share the same cathode connection. Be sure yours is also a common cathode display, and not a common anode display.

Finally, here’s one more example to close out our discussion. Refer to **Figure 8** on how you can

See Figure 2 for other wiring connections



connect a 74595 SIPO chip to a seven-segment display. Use the **Listing 6** sketch to send out eight bits to light the seven segments plus the decimal point.

With this setup, you’re still able to produce all 10 digits, of course, plus Predator-type alien language symbols that only you (and your robot) understand. **SV**

Listing 6.

```
int latchPin = 8;
int clockPin = 12;
int dataPin = 11;

void setup() {
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
}

void loop() {
  // Write a regular 7
  digitalWrite(latchPin, LOW);
  shiftOut(dataPin, clockPin, LSBFIRST, B11100000);
  digitalWrite(latchPin, HIGH);
  delay(1000);
  // Write some funky character
  digitalWrite(latchPin, LOW);
  shiftOut(dataPin, clockPin, LSBFIRST, B01110011);
  digitalWrite(latchPin, HIGH);
  delay(1000);
}
```



Continued from page 19

- Altera Cyclone® III FPGA.
- Altera Nios® II embedded processor.
- Software CD with designing/implementing tools.
- A 137 page lab manual by university professor, Enoch Hwang, PhD.
- iPad version of the lab manual available on iTunes.
- Eleven hands-on labs correlated to any textbook for microprocessor design training.
- Works with any Windows XP or higher system (32-bit only).
- Rugged, lightweight, blow-molded carrying case.
- Complete kit with 100 machined pin hookup wires.
- Sixteen LEDs.
- Three seven-segment displays.
- Sixteen slide switches and three pushbutton switches.
- Expandable breadboard allows the system to grow as knowledge increases.
- Perform experiments using standard TTL logic.
- Four each, regulated 5V power (Vcc) and ground (GND) points.
- Eight input/output connection sockets.
- High quality machined sockets for input and output interconnections.
- The DL-030 can implement all of the experiments and circuits of the DL-010 and DL-020 trainers plus more.

For further information, please contact:

Global Specialties

Website: www.globalspecialties.com

Bluetooth Module for Streaming Audio

Microchip Technology, Inc., has announced the expansion of its wireless product portfolio with a certified Bluetooth® audio module that supports audio for voice and music.

The RN52 module provides extremely low power consumption in a small, surface-mount form factor, and includes standard Bluetooth audio and data profiles for all smartphone platforms. These features help designers add high quality wireless audio — combined with data capabilities — in a broad range of applications such as wireless stereo speakers, headphones, automotive hands-free, medical devices, and computer accessories.

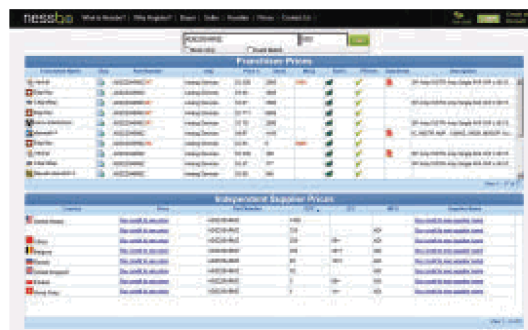
The RN52's embedded Bluetooth stack includes the popular SPP, A2DP, HFP/HSP, and AVRCP profiles, along with the iAP for use with an iPhone® or iPod®. Additionally, the RN52 supports a variety of audio codecs such as SBC, aptX®, AAC, and MP3. A Bluetooth audio evaluation kit is available for \$169.95.

For further information, please contact:

Microchip Technology, Inc.

Website: www.microchip.com

Search Engine For Electronic Components



Nessbo.com is a new search engine for electronic components that hopes to set a higher standard for simplicity, versatility, and scope. "As far as I know, www.nessbo.com is the first search engine that combines every imaginable source of parts, displays prices from all of them, and lets purchasing managers buy directly from the source by a click of a button. It's all on one screen and extremely easy to use," stated company CEO Shimon Tiran.

Nessbo.com has been in intensive development for the past two and a half years, and has been in full-scale beta testing since November of last year. The search engine — which launched officially on January 10th — offers free searches for franchisee parts. Among those franchised distributors are Avnet, Arrow, Digi-Key, Mouser, Farnell, Future, RS, and all of their subsidiaries, plus at least 30 others. An optional feature that isn't available on any other website, according to Tiran, is that any registered user can take advantage of the optional built-in Customer Relationship Management (CRM) system. The CRM lets users keep track of all searches ever made on the site, and allows the searches to be assigned to particular projects, clients,

employees, or any other category. There is a nominal charge for deeper searches of independent suppliers, with package prices starting at \$35.

It costs nothing to register with **Nessbo.com**, and even paid subscribers are guaranteed an automatic refund if they aren't happy with their search results. "All they have to do is click the 'Refund' button. We don't ask any questions," Tiran said.

For further details, please contact:

Nessbo.com



ROBOTSHOP WILL INTRODUCE...

— SERVO ERECTOR SET V1.1 —

05.24.2013

GET READY TO CHANGE THE WORLD OF ROBOTICS



THE COUNTDOWN HAS BEGUN

Lynxmotion is a trademark of
RobotShop Distribution inc.

RobotShop.com
Robotics at Your Service!™



Build the Kronos Flyer

Part 5: Upgrades



by Michael Simpson

www.servomagazine.com/index.php?/magazine/article/march2013_Simpson
Discuss this article in the SERVO Magazine forums at <http://forum.servomagazine.com>

With guidance from the previous articles in this series, you have assembled, configured, and flown your very own Kronos Flyer quadcopter. This particular platform — our reference build — was designed in a way to minimize the cost of repairs when you crash (you did, right?), to learn on, to make mistakes with, and to hone your skills. One of the advantages to building your own multi-rotor craft is the opportunity to modify it and add functional upgrades. This article will show you some upgrade possibilities — not as a step-by-step guide, but as a germinator to inspire your imagination and to encourage your own add-on design ideas.

Upgrades

Boom Extensions

I have designed and assembled no less than 40 variations of the Flyer. During these design iterations, I have been able to create a somewhat modular frame design. For instance, the aluminum booms can be used on the 430 mm reference build, or on an extended 560 mm platform.

To create the 560 mm booms, I designed a set of extensions that attach to them. These boom extensions allow the motors to hang off the ends of each of the booms as shown in **Figure 1**. These extend the motors out an additional 65 mm on each side.

Extending the booms this way will present more motor and prop options at very little cost in weight. These extensions will allow you to increase the prop size to 9" when using the original Flyer reference motors. Expect flight times to be around five minutes when using the 8 x 4.5 slow fly props and a three-cell 2,200 mAh battery.

I also created a set of extensions that can be used with the popular hexTronik DT700 and DT750 motors available from HobbyKing. These motors are very efficient and offer longer flight times when utilizing larger props.

When using the DT700 motor, I recommend a 10 x 4.5 prop and three-cell battery. I have achieved seven minutes of flight time using the DT700 motors and a three-cell 2,200 mAh battery with this configuration, and flight times of 12 minutes when using a three-cell 3,300 mAh battery.



FIGURE 1.

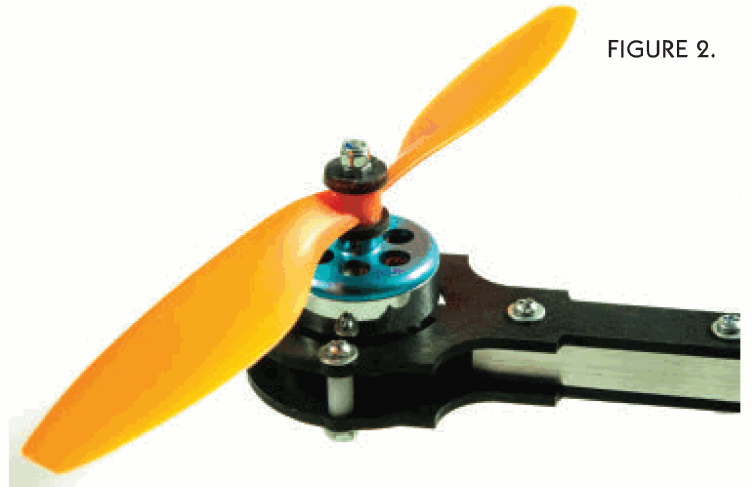


FIGURE 2.

The DT7xx extensions shown in **Figure 2** have a larger cutout and a different hole pattern for the motor mount.

Once you have added the boom extensions, your craft will look much like the one in **Figure 3**. You will also find that your craft won't be as agile as it was with the shorter booms. It will however, be more stable.



FIGURE 3.



FIGURE 4.

Lighting

Before moving on to the platforms, I want to talk a little about lighting. This is an optional upgrade. However, it is a must if you plan on flying your craft in the late evenings or at night. Proper lighting like that shown in **Figure 4** will make it easier to tell where you craft is and what direction it is facing. You also have to take into account the "cool" factor.

For the lighting, I used LED strips like the one shown in **Figure 5**. The strip is 1/4" wide and contains a colored LED every 3/4" inch

You have a couple of ways to control your LED strips. They are designed to be operated from a 12 volt source, so you can connect them directly to your main battery by attaching them to your power distribution board. You can also use an RC controller like the ones shown in **Figures 7** and **8**.

The receiver controlled switch shown in **Figure 7** will allow you to turn all your lights on and off by using a free channel on your receiver. The multi-remote switch shown in **Figure 8** will allow you to control up to seven individual LED strips.

I prefer the multi-remote switch, shown again in **Figure 9**. It comes with all the connectors needed to attach your strips and power. By using a single switch on your radio, you can cycle through the strips. By toggling the switch, your neighbors will think they're seeing a UFO.

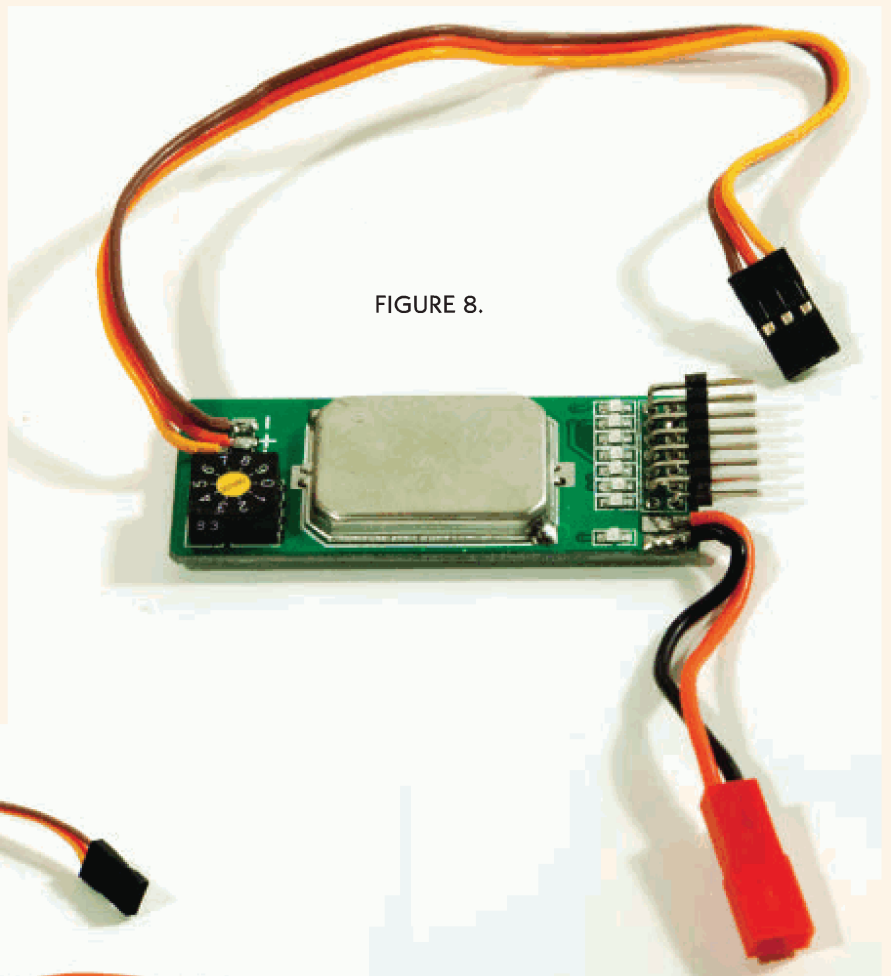


FIGURE 8.

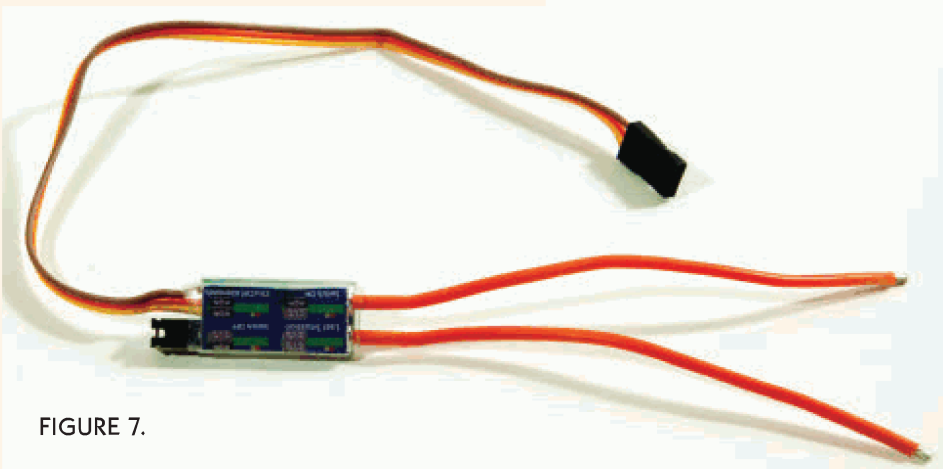


FIGURE 7.

or so. This particular strip was purchased from HobbyKing; you may be able to purchase some from your local hobby store.

The LED strip has adhesive that

will allow you to attach it to your booms. One end of the strip has a couple of wires attached. I soldered a small two-pin header to the wires and then attached them to a cable that

came with my lighting controller. The wires are routed up the boom and held in place with three tie wraps as shown in **Figure 6**.

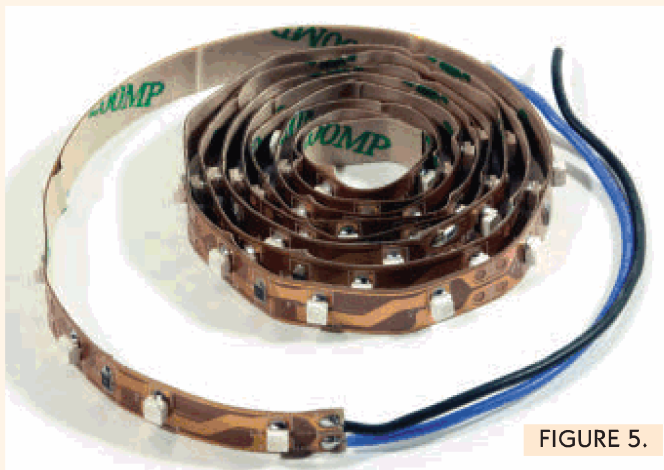


FIGURE 5.

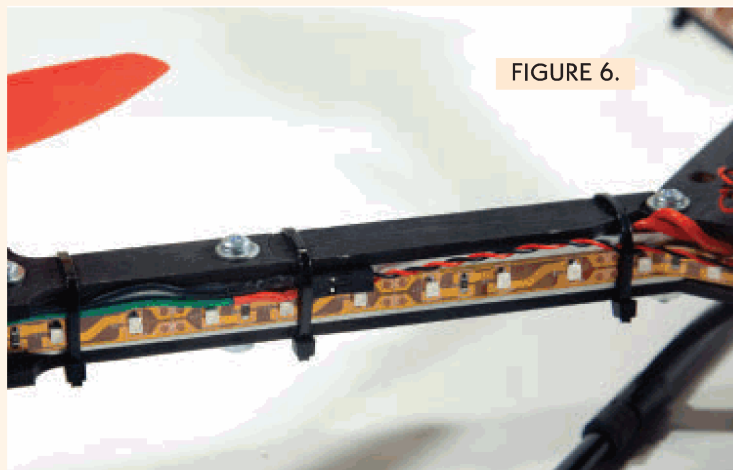


FIGURE 6.

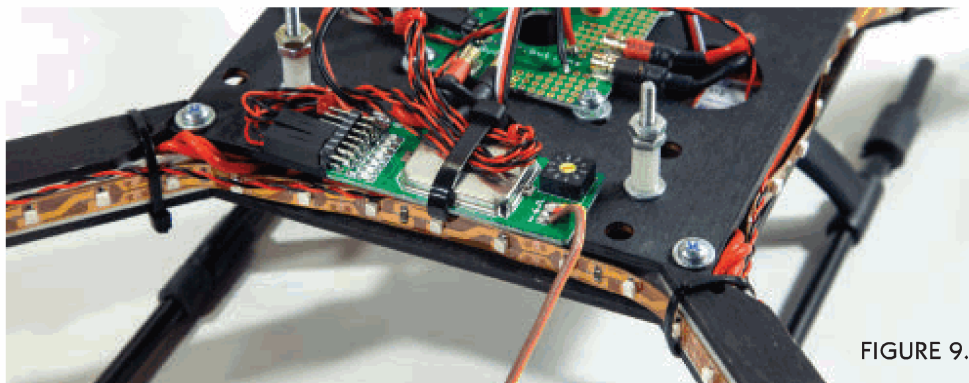


FIGURE 9.

I decided to do the same with the various component platforms I have designed. You are already familiar with one of these platforms. The reference platform shown in **Figure 10** is the one you have currently attached to your Flyer.

This reference platform is fine for learning to fly your quadcopter, but what if you want to add, say, a camera? Let's look at what it would take to add the GoPro camera shown in **Figure 11** to your beast.

Platforms

In last month's article, I mentioned the UNM. This is a set of small platform pieces that allow you to easily move your NAZA, receiver, and GPS between models. It also allows you to quickly remove those components from your craft for maintenance or transport.

At the very beginning, you are presented with two problems. First, there is not enough space to easily mount the GoPro and one of its cases to our basic platform. The other problem is vibrations. Multi-rotors are notorious vibration generators, and these vibrations do not play well with cameras.



FIGURE 11.

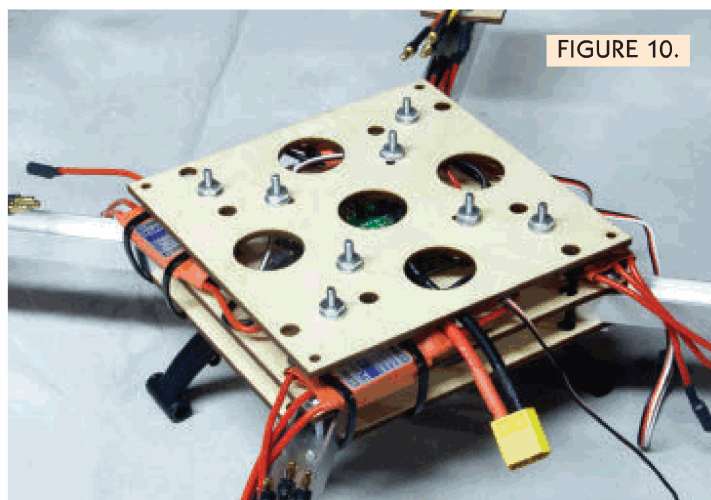


FIGURE 10.



FIGURE 12.

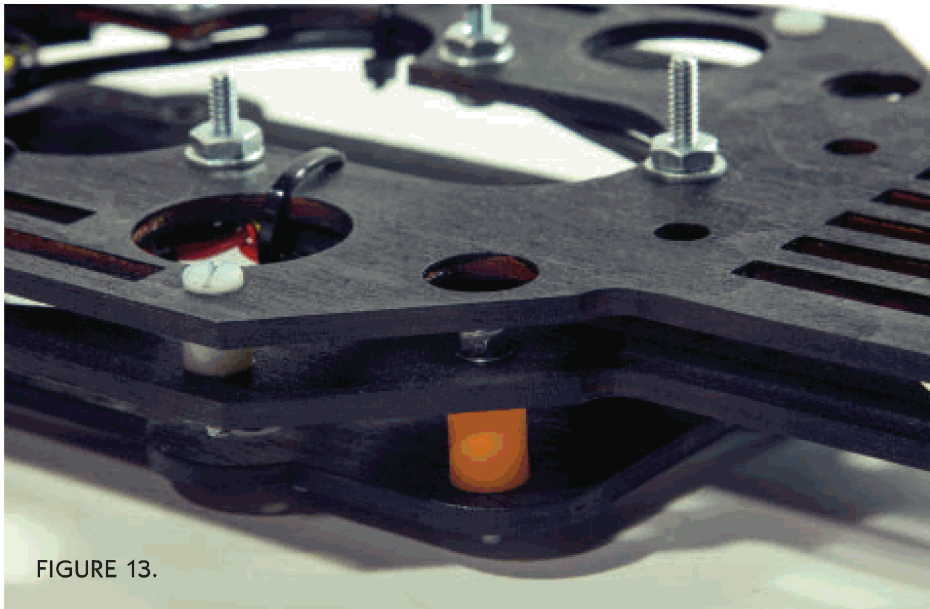


FIGURE 13.

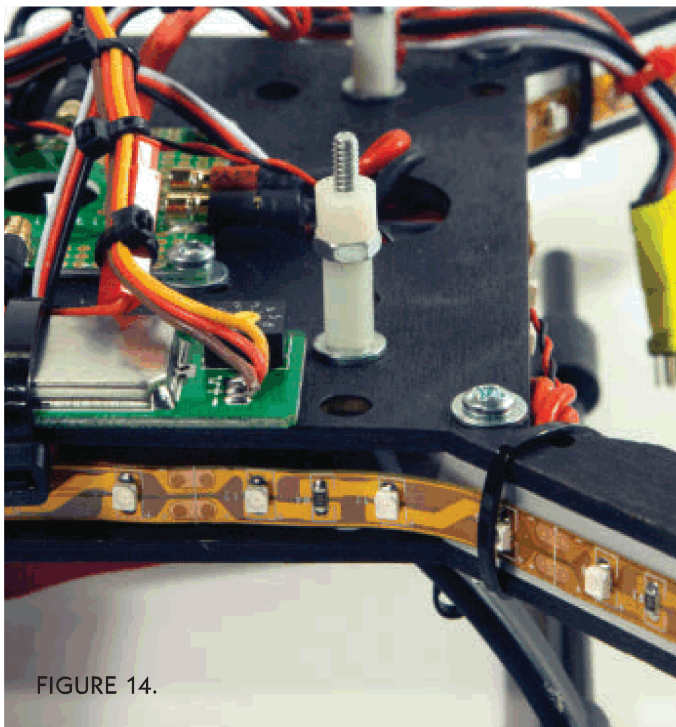


FIGURE 14.

frame to hold servos or other hardware that's not appropriate for the isolated platform (clean frame).

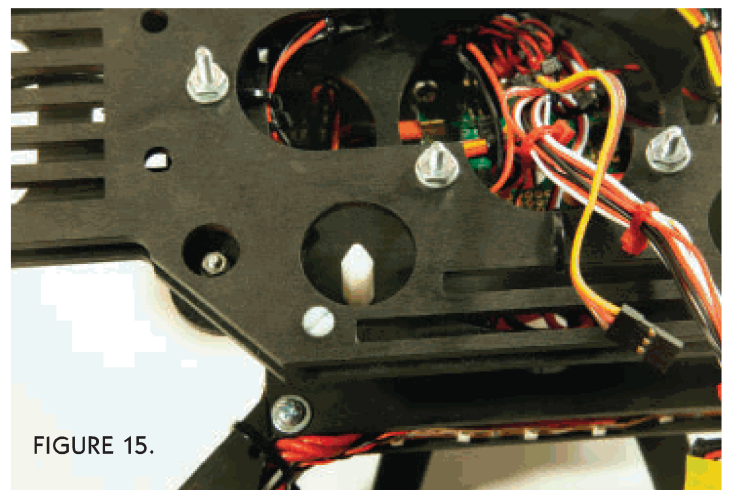


FIGURE 15.

You need a platform that is both larger and vibration free. The multi-layer platform shown in **Figure 12** fits the bill perfectly. It is large enough to hold two cameras, a video transmitter, and an OSD (object-based storage device) interface — all with room to spare.

The platform shown here already has all the wiring for the above tucked between the top two layers. These top two layers are isolated from the lower layer by using small elastic bobbins as shown in **Figure 13**. This keeps the upper two layers vibration free.

To install the new platform, first remove the original platform. Next, add four 1/4" spacers as shown in **Figure 14**. Lower the new platform over the screws. It is held in place with four 1" standoffs shown in **Figure 15**. The standoffs are attached to the non-isolated portion of the frame (dirty frame).

If longer extensions are used, it is possible to add an additional

Once the platform is attached to the frame, add the controller assembly (UNM) to the platform as shown in **Figure 16**.

Attach Camera

It is a simple matter to connect the GoPro to the platform using one of the slits. These slits are perfect for positioning tie wraps to hold the mount in place; take a look at **Figure 17**. While this setup works fine, the GoPro case and mount add over 90 grams to the weight of the craft. In addition, you will need to purchase a special skeleton case if you plan on making connections to your GoPro.

I created my own skeleton case (**Figure 18**) that connects directly to the platform. It allows cable connections to the GoPro, and weighs only 20 grams.

With the camera securely in place, you are now free to make onboard videos of your flights. Please note that if you run the camera in wide mode, the props will be in the shot. Set the view to medium or normal if you want a clear shot.

Video Transmitter

If you really want to take your Flyer to the next level, add a video transmitter to your platform. **Figure 19** shows an Immersion 5.8 GHz transmitter. The transmitter is strapped to the back of the platform using a tie wrap. It's connected via a small adapter cable to the GoPro 2 camera. All that is needed to transmit your flight is to power up the Flyer and turn on the camera.



FIGURE 16.

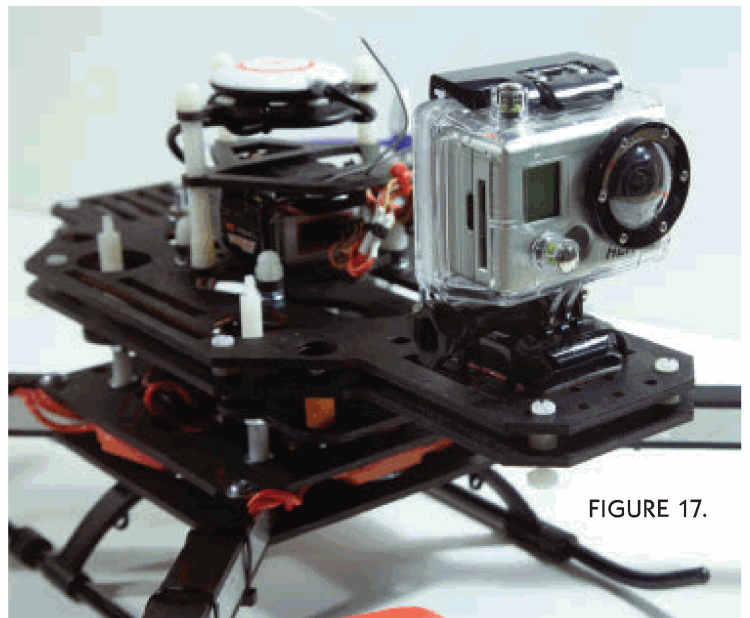


FIGURE 17.

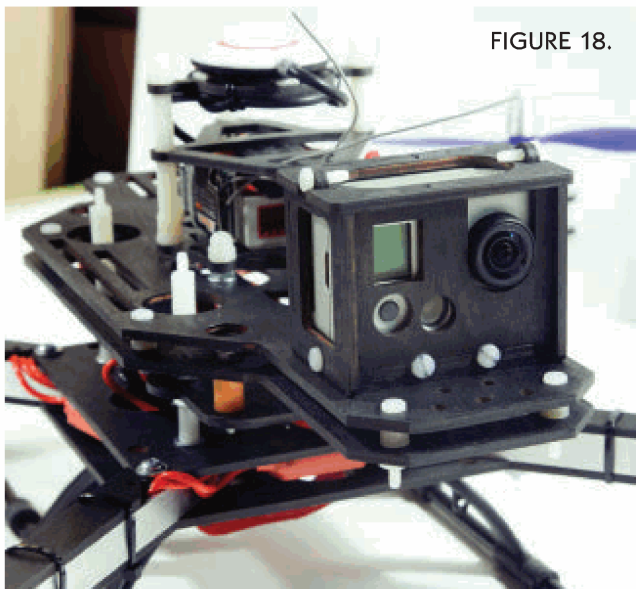


FIGURE 18.



FIGURE 19.

Ground Station

A ground station consists of a receiver, display, and antenna. The Fat Shark Dominator goggles shown in **Figure 20** are a self-contained ground station. They have a built-in 5.8 GHz receiver and a circular polarized antenna. They work very well as long as you don't wear glasses. You can purchase a myOptic Lens set, but I found they just don't yield a clear image when using the MyOptic lenses.

SparkFun sells a 7" monitor that works perfect as an outdoor display. To receive the video signal, I used an



FIGURE 20.



FIGURE 21.

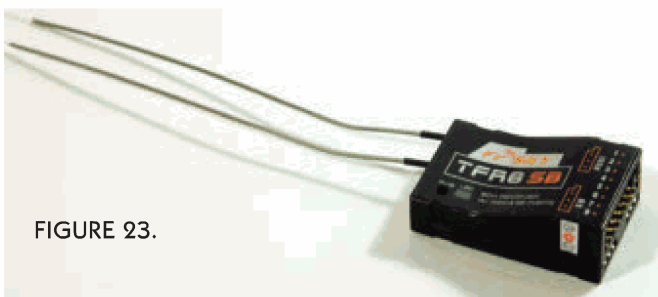


FIGURE 23.

Immersion 5.8 GHz receiver shown in **Figure 21**.

To create a ground station that I could hold in my hand, I mounted the monitor, receiver, and battery to a panel and attached it to the handle on my Futaba transmitter as shown in **Figure 22**. Being a little heavy, I used the neck strap that came with the radio for additional support on extended flights. A three-cell 2,200 mAh battery will power the station for hours.

Futaba Radio Upgrade

If you are serious about getting into aerial robotics, I can't stress enough how important it is to purchase a Futaba 8FG radio system. As you add functionality to your craft, you will need more channels. The FrSky TFR8SB receiver shown in **Figure 23** has the ability to add seven more channels to your craft. Simply place a jumper on channel 1, and the remaining channels will be mapped to channels 9-14 on your transmitter.

APM 2.5+ System

At the request of a few readers, I decided to pick up an APM 2.5+ system. The APM (ArduPilot Mega) is a flight controller that promises great functionality in the aerial robotics arena. The controller has a ton of accessories that can be used to add functionality. **Figure 24** shows some of



FIGURE 22.

these accessories, which include:

- APM 2.5+ assembled and tested flight controller, GPS, current monitor.
- GPS upgrade.
- Minimal OSD module.
- 433 MHz two-way telemetry module.

How did it work?

I built a Flyer specifically for testing this system. There is a ton of information on the ArduPilot website (www.diydrones.com), but I need to point out that much of it is outdated, so it's hard to sift through the parts that apply to the APM 2.5. Once I located the correct web pages, however, I was able to get the controller, radio, and GPS installed.

It flew well. The two-way telemetry worked great and I was able to track my progress in real time — even through my crashes. Stability became a problem, however, after several flights, when the flyer would simply fall out of the sky. Signal range wasn't the problem as I was standing only a couple feet away during several of the drops. For that reason, I feel the APM has great potential, but I would not trust it to fly my \$4,000 camera around — at least until some of the kinks are worked out of the software.

That being said, I did an experiment. I added the system

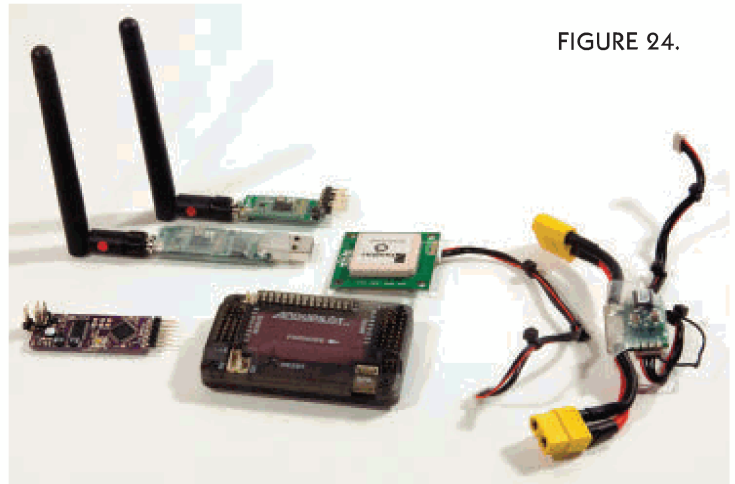


FIGURE 24.

to one of my NAZA controlled craft and used the APM as an advanced telemetry unit. While I used the NAZA to fly my craft, the APM flawlessly reported GPS information to my laptop. I am going to experiment with this more, as I have high hopes of using the APM to control a multi-axis camera gimbal via remote control.

Was it as stable as the NAZA? No. Nowhere close. Was it better than the KK2 board? A hands down yes. FYI, I was never able to get the OSD module to work, even after hours and hours of research and attempts.

Conclusion

Before closing, I'd like to shed some light on a couple of available frames that are interesting. The frame shown in **Figure 25** — by **HoverThings.com** — is a great FPV (First Person View) frame. While it does not have an isolation platform, I am told they are working on one. I consider this to be a middle of the road frame with a lot of potential. The second frame from **FPVmanuals.com** shown in **Figure 26** is the cream of the crop, and the most expensive. It sports aluminum arms and an isolation platform. This is the frame that all others try to copy.

While I would stick with a square frame for aerial robotics, these two frames are at the top of their game when it comes to FPV flying. Be sure to check them out. Notice the UNM (Universal NAZA Mount) on each craft. I can move my NAZA, GPS, and receiver from craft to craft without changing a single setting.

This has been both a fun and exhausting series. Much of the research will continue on my website at www.kronosrobotics.com. Be sure to visit often, as I will be posting the sources for all the products mentioned in this article along with videos. You can also check things out at the article link for updates. Please post any questions you may have to the *SERVO Magazine* forums at <http://forum.servomagazine.com/viewtopic.php?f=49&t=16866>. **SV**

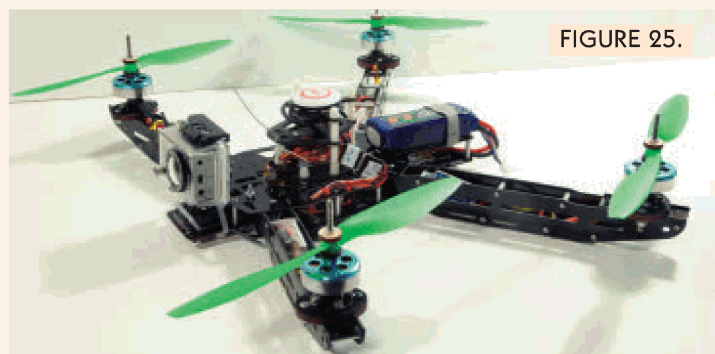


FIGURE 25.

One More Thing: This was to be the last article in this series, however, I have received several requests to look deeper into the APM 2.5+. I have completed the setup of the APM 2.5+ and updated to its latest firmware. I say it's time we automate our Flyer and had it run some waypoints. Next month, I will dive into the flight tests of the ArduPilotMega and its latest firmware.

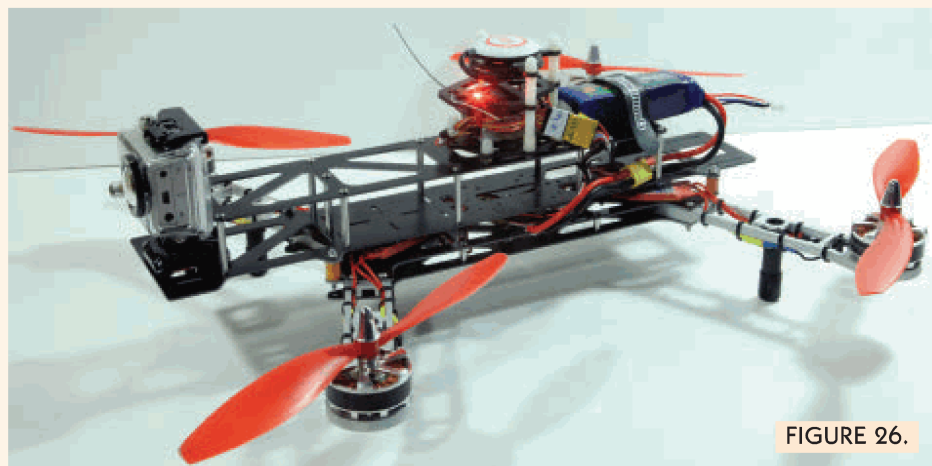


FIGURE 26.

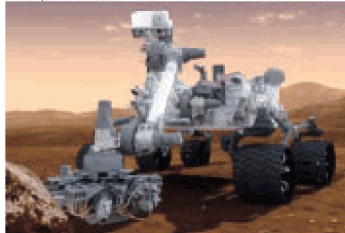
TCF™-2013

The College of New Jersey Hosts
The 38th Anniversary of
THE ORIGINAL PERSONAL COMPUTER SHOW

38th Annual

Trenton Computer Festival™

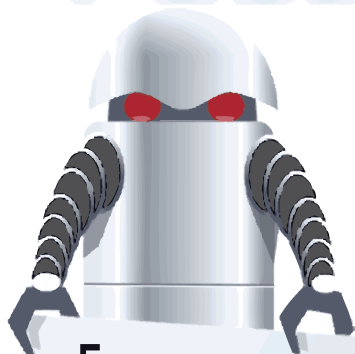
NASA/JPL-Caltech



FIRST® Robotics Competition

FIRST Mid-Atlantic Robotics District Competition:

TCF Attendees are invited to visit a FIRST high school robotics competition being held Friday and Saturday (March 15-16) in the TCNJ Student Recreation Center. Admission to FIRST is free; food is available. Seeding rounds will run continuously 9:00am to 5:00pm on Friday and until noon on Saturday. Elimination rounds follow lunch on Saturday.



Focus on
ROBOTICS
at TCF'13

Admission:

One-Day Ticket (at gate) - \$10.00

Rain or Shine - Free Parking

www.tcf-nj.org

**@ The College of New Jersey, Ewing, NJ
Saturday, March 16, 2013 (One Day Only)**

9:00 am — 5:00 pm

Registration & Flea Market open at 9am, Talks start at 10am.

• Apple/Mac/iOS Day

Full Day with Apple's Dave Marra

• **Windows 8 Presentation by David Soll**

• **Object-Oriented Programming University**

**50+ Talks, Workshops, Tutorials,
Demos and Special Events!**

TALKS ON WINDOWS, MAC and LINUX.

**The Flea Market and Vendor Fair will
all be INDOORS, along with the talks.**

Special TCF room rates at the Marriott
Courtyard Ewing-Hopewell Hotel in
Ewing Twp.; reserve a room in the
TCF/ITPC block at (609) 771-8100.

**DOOR
PRIZES**

**MULTIMEDIA, 3D PRINTERS, PHOTOGRAPHY
HOME CONTROL, SECURITY and WIFI**

**Get an Amateur Radio License at TCF!
Ham Cram Session & Exam**

Arduino Developer/User Tutorial/Workshop

HackTCNJ Hackathon, from noon on March 16 to 2pm on the 17th:
Calling all students in the greater NYC/NJ/Philly/DC corridor. Come hack your best app and compete for fame, glory and prizes at HackTCNJ 2013! Located on the campus of The College of New Jersey, in Ewing, NJ, this hackathon welcomes students (both college and high school) for a friendly app-building competition. For Info: <http://hacktcnj.com>

TCF IT PROFESSIONAL CONFERENCE**

Friday March 15, 2013, 8:30am — 5pm

For Conference Info and Fees See

<http://www.tcf-nj.org/pc>

**For additional TCF'13 info,
directions and advance tickets:
www.tcf-nj.org**

****The IT Pro and ISEC Conferences require separate registrations.**

Integrated STEM Education Conference (ISEC)**

**Saturday, March 9 – 9:00am—5:00pm
at Friend Center, Princeton University, NJ**

Theme: STEAMing Synergies

Discussion and exploration of best practices
in K-20 STEM education (extra cost registration).

Continuing education credits available!

Info: <http://ewh.ieee.org/conf/stem/>

The 2013 Trenton Computer Festival™ is sponsored by The College of New Jersey and its School of Engineering.

DYNAMIXEL PRO

The Solution for Full Scaled Robots

High Power, High Precision

Up to 200W BLDC motor and up to 500,000 pulse per turn (4096 absolute resolution)

Full Modular Solution

Ease of implementation (no additional design for parts needed)
Ease of maintenance (simple part swapping)

Sophisticated Control Algorithms

Position and speed input commands
with dual-loop current control

Novel Gear Reduction System

High torque output, light weight,
with high impact resistance



Data	H42-20-S300-R		H54-100-S500-R		H54-200-S500-R	
	Unit	Data	Unit	Data	Unit	Data
Rated voltage	V	24	V	24	V	24
No load speed	RPM	28.3	RPM	35.2	RPM	35
No load current	A	0.61	A	1.06	A	1.18
Continuous operation	Speed	RPM	RPM	32.7	RPM	32.1
	Torque	Nm	Nm	21.142	Nm	39.131
Maximum output power	W	23.64	W	144.58	W	262.66
Resolution	Step/turn	304,000	Step/turn	502,000	Step/turn	502,000
Gear ratio	-	304	-	502	-	502
Backlash	arcmin	3.5	arcmin	3.5	arcmin	3.8
Interface	-	RS-485 / CAN	-	RS-485 / CAN	-	RS-485 / CAN
Operating temperature	°C	5~55	°C	5~55	°C	5~55

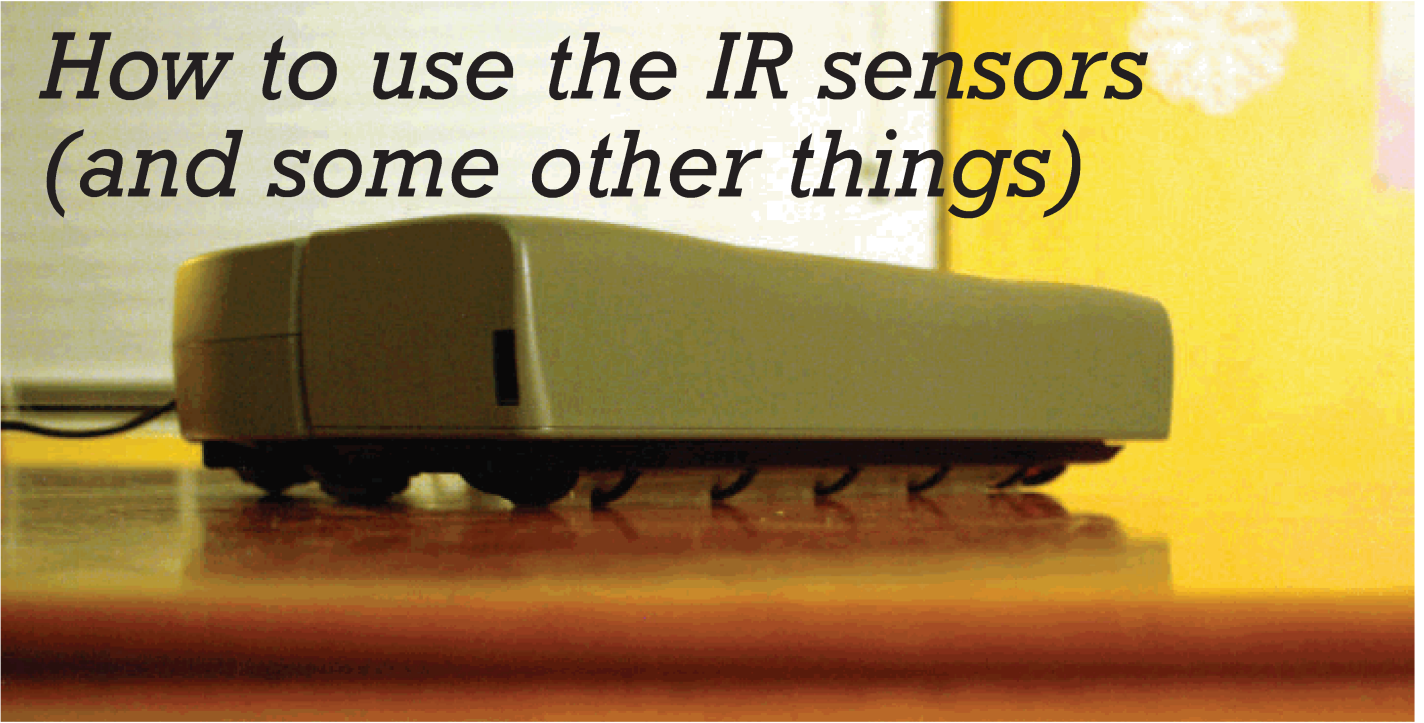


DARWIN-OP



The Neato Robot and ROS

How to use the IR sensors (and some other things)



by Boris Kudryavtsev
Junior High (8th grade)

Have you ever imagined a vacuuming robot driving on a table? Probably not ... Well, it is now possible with the use of ROS (Robot Operating System), developed by Willow Garage — a small, but growing company located in Menlo Park, CA. ROS provides libraries and tools to help software developers create robot applications. It offers hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and more. This article will teach you how to use this tool to manipulate the Neato™ automatic vacuum cleaner.

For those unfamiliar with the Neato, it is a robotic vacuum cleaner developed by Neato Robotics. The device is very hacker-friendly. It is assembled with normal ("+" type) screws, and the software is easily accessible through the USB interface.

The machine is famous for its low cost LIDAR (Light Detection And Ranging) scanner, which is basically a spinning laser which collects distances from the objects around it. This gives the robot the ability to make its way through a house efficiently while it is vacuuming.

The robot also has a pair of IR (infrared) distance sensors which prevent the device from falling off edges and going down stairs. There is an accelerometer and one more IR sensor to detect walls.



Figure 1.

For the hack I'll discuss here, you will need the following things:

- Neato XV-11 (or any other in its family, though I do not guarantee it will work) .
- A laptop (or netbook) running Ubuntu Linux (versions 10.04 up to 11.10; I recommend 10.04).
- A mini-B USB cord (preferably as long as possible).

Let's Get Started

Step 1: Install ROS Electric.

If you have already done so, proceed to Step 2.

Follow the installation instructions on www.ros.org/wiki/electric/Installation/Ubuntu. Because of the new release of ROS, the tutorials were modified. So, follow the instructions on the website listed in the **Resources** section. If you don't have any experience in ROS whatsoever, continue through the tutorials provided on the website.

Step 2: Getting the Driver Package.

First, copy the SVN directory *neato_robot* to your *ros_workspace* folder:

```
cd ~/ros_workspace  
svn co http://albany-ros-pkg.googlecode.com/svn/trunk/neato\_robot
```

(This package was written by Michael Ferguson.)

Step 3: Connecting the Neato.

Connect the Neato to the computer with the mini-B USB cord. You will find the input for the cord under a small rubber flap on the left rear of the robot beside the charging jack (**Figure 1**). Then, run:

```
lsusb  
Find the Neato™ device (mine was ID 2108:780b), then:  
  
sudo modprobe usbserial vendor=0x2108  
product=0x780b
```

You should now have a `/dev/ttyUSBX` like device connected. (X is a number indicating the device, i.e., 0.)
Now, run:

```
roslaunch neato_node bringup.launch
```

If all is successful, this will start the node; you will hear the LIDAR spin up, and at the end of the terminal's text you will see a similar inscription:

```
[INFO] [WallTime: 1358295910.147500]  
Using port: /dev/ttyUSB0
```

Troubleshoot: If you get a long red error script, it might say something like this at the end:

```
/dev/ttyUSB0 [No such file or directory]
```

Try reconnecting and repeating Step 3. If that does not fix the problem, try this:

`cd` to `~/ros_workspace/neato_robot/neato_node/nodes`, and open `neato.py` using `gedit` (or any other text editor; for instance, `vi` or `vim`).

Go down into the class `NeatoNode` where there should be a line which starts with `self.port`; change: `/dev/ttyUSB0` to `/dev/ttyACM0`.

If that doesn't do the trick, try rebooting. After you have done so, repeat Step 3.

Step 4: Reading sensor values.

Download the `neato.py` file from the article link, then go to `~/ros_workspace/neato_robot/nodes/` and replace the `neato.py` file with the one you downloaded.

After starting the node and running the `rostopic` list, you should now see a list of topics similar to this:

```
/cmd_vel  
/drop_l  
/drop_r  
/wall  
/odom  
/laser_scan
```

If you run `rostopic echo drop_l`, you should get a series of values (in meters) showing the distance that is read off the [robot's] left bottom IR sensor, similar to this:

```
header:  
  seq: 1  
  stamp:  
    secs: 1358296029  
    nsecs: 564053058  
  frame_id: base_link  
radiation_type: 1  
field_of_view: 0.0175438597798  
min_range: 0.0599999986589  
max_range: 0.15000000596  
range: 0.0599999986589  
---
```

It would be the same for the `/drop_r` and `/wall` topics.

Step 5: Turning it into a TABLEBot.

For years, the HomeBrew Robotics Club of Silicon Valley, CA has hosted an event called the TABLEBot Challenge. A TABLEBot is a robot designed to survive, live, and play on a table. This challenge is meant to resemble tabletop soccer but is done in "Phases" for accessibility:

Phase I: Build a robot that goes from one end of a table to the other and back.

Phase II: Have the robot push a block off the edge of the table.

Phase III: Have the robot push the block into a shoebox mounted at the end of the table.

We will make the Neato do the second phase, in which it should find a block on a table and push it off the edge without falling off with it.

Resources

ROS Electric installation instructions:
<http://mediabox.grasp.upenn.edu/ros/wiki/ROS%282f%29Tutorials%282f%29InstallingandConfiguringROSEnvironment.html>

Willow Garage:
www.willowgarage.com

ROS:
www.ros.org

Neato Robotics:
www.neatorobotics.com

Michael Ferguson:
<http://www.ros.org/wiki/MichaelFerguson>

Note: You can find the SVN directory used in this article by clicking the albanys-ros-pkg link and then `neato_robot`.

ROS page for `neato_robot`:
www.ros.org/wiki/neato_robot

Note: For some reason, someone has removed the SVN directory from this page.

Homebrew Robotics Club of Silicon Valley
www.hbrobotics.com

Figure 2.



Create a Package

First, create an ROS package named *table_neato* with the dependencies: *geometry_msgs* (movement control), *sensor_msgs* (sensors!), and *roscpp* (ROS c++ libraries). The *neato_node* dependency is optional:

```
roscatkin pkg table_neato geometry_msgs sensor_msgs roscpp neato_node
```

Second, download the *phase2.cpp* from this article's link and place it into the *src* folder of the package we just created.

IMPORTANT: Download the *neato_driver.py* file as well, and place it into the *neato_driver* folder inside the *neato_robot* package. There are a couple of modifications in that program that make the motors more responsive.

After the robot had spotted its target — to be careful — stand beside the table and be ready to catch the robot in case something unexpected happens.

Now, open the file in *table_neato* called *CMakeLists.txt* and add this at the end:

```
roscatkin add_executable(phase2 src/phase2.cpp)
```

Launching the Program

Now, place the robot on a table, as well as some

object that is taller than four inches and is durable enough to withstand a fall of four feet.

After you have done all of that, you just have to compile and run the code, which is done like this:

```
cd to table_neato,
make
```

If everything is well, it should compile. Start the node:

```
roslaunch neato_node bringup.launch
```

Hold the laptop and be sure to stand back away from the table, keeping the cord down as low as possible since it might interfere with the LIDAR scan.

Open a new tab (or new terminal window) and run the code:

```
roscatkin run table_neato phase2
```

(If you press TAB after *table_neato*, the rest would be auto-filled. If you're new to Linux, that's a nice little feature in the terminal that you can use with any command to save time. It's especially useful when you have to enter long directories.)

The robot should now complete the challenge (**Figure 2**).

If you are confident, you can always improve my code and/or write new code for the device.

There is still the third and final challenge, in which a robot should find a block and push, carry, or throw it into a shoebox mounted at the end of the table. You should think about participating. **SV**

The aspiration of many robot hobbyists is to create a machine that uses autonomous behavior to achieve a goal. Unfortunately, many roboticists assume that such goals are limited to simple tasks like avoiding objects or following a line. This article discusses how giving your robot even a rudimentary ability to identify objects can increase its ability to cope with more complex situations.

Robotic Sensors

Part 1: Identify and Distinguish

by John Blankenship and Samuel Mishal

www.servomagazine.com/index.php?/magazine/article/march2013_Blankenship
Discuss this article in the SERVO Magazine forums at <http://forum.servomagazine.com>

A typical hobby-level robot might avoid objects while it randomly roams around a room. Nearly all hobbyists have observed such a behavior, and only a novice will be overly impressed by the capabilities of such a robot. In actuality, the robot's random movement is the source of discontent. The robot's only goal is the avoidance of *anything* that gets in its way. If the robot could identify important objects, then its ability to react to its environment could be greatly improved.

If, for example, the robot could identify both a soccer ball and a goal, then it could be programmed to find the ball and deliver it to that goal. The task of avoiding other objects during this endeavor then becomes a means

to an end, rather than a goal of its own.

Programming a robot to actually play soccer can be very ambitious, but a simplified version of this activity can help demonstrate how even just a hobby robot can identify objects within its environment.

The Needed Sensors

In order to achieve this task, our robot will need to have several types of sensors. The details of the algorithm we develop will depend heavily on the actual sensors used. So, we will use a modular approach to make it easy to implement the principles discussed here with sensors you might already have.

First, our robot will need a

ranging sensor capable of measuring the distance to objects. We'll use an ultrasonic sensor mounted with a forward orientation. The purpose of this sensor will be to locate objects that are reasonably close to the robot.

The robot will also require a beacon detector to easily identify the location of a specified target (a beacon will be mounted above the soccer goal). We will use a RobotBASIC IR beacon, but you could use a flashing light or a uniquely colored object. (A beacon can be almost anything, as long as you have a sensor that can detect it.)

We'll mount a U-shaped trap to the front of the robot as shown in **Figure 1**. This trap is made from craft sticks (like large Popsicle sticks) that make it easy for the robot to trap the ball and push it around.

FIGURE 1.

The trap has three sensors of its own to help identify and maintain control of the ball. Our trap sensors are digital IR versions (Pololu #1134) with a fixed range of about four inches. Two are mounted on the tips of the trap – pointing forward – with the third mounted on the side pointing inward (refer again to **Figure 1**). We will see how these sensors are used shortly.

Figure 1 also shows a Parallax Ping))) ranging sensor (part #28015) mounted on the very front of the robot, as well as a beacon detector which is a Vishay TSOP34156 with a short cardboard tube taped to the front of it to make it more directional.

Our robot is a Magician chassis (available from Zagros Robotics) controlled by the RobotBASIC Robot Operating System (RROS), along with a special printed circuit board (PCB) to make it easy to directly interface with all the sensors.

Once you understand the principles we're using here, you can create a similar project with almost any hobby robot equipped with appropriate sensors.

Most sensors supported by the RROS plug directly into the PCB. As mentioned earlier, our soccer robot needs three sensors mounted on the ball trap. **Figure 2** shows how easily this can be done. We simply drill the appropriate holes for the IR sensors, then hold them in place with the female end of the cables that connect the sensors to the line sensor inputs on the PCB.

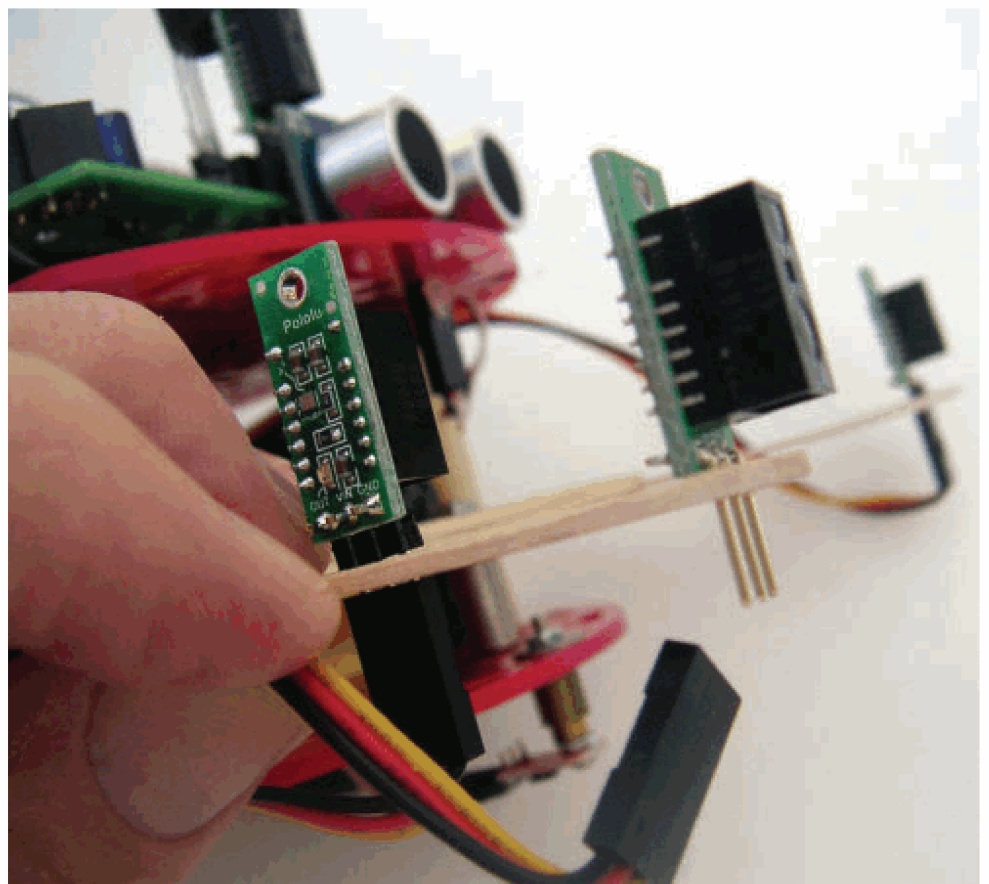
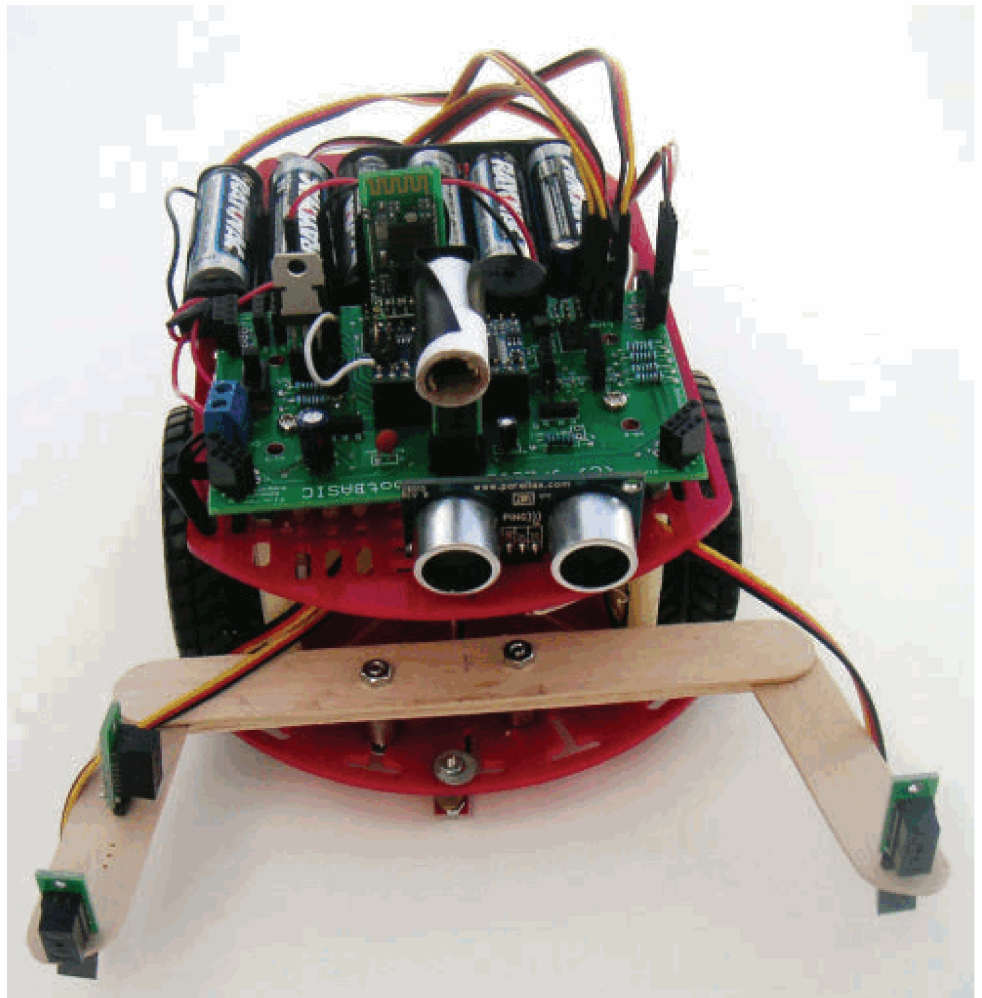


FIGURE 2.

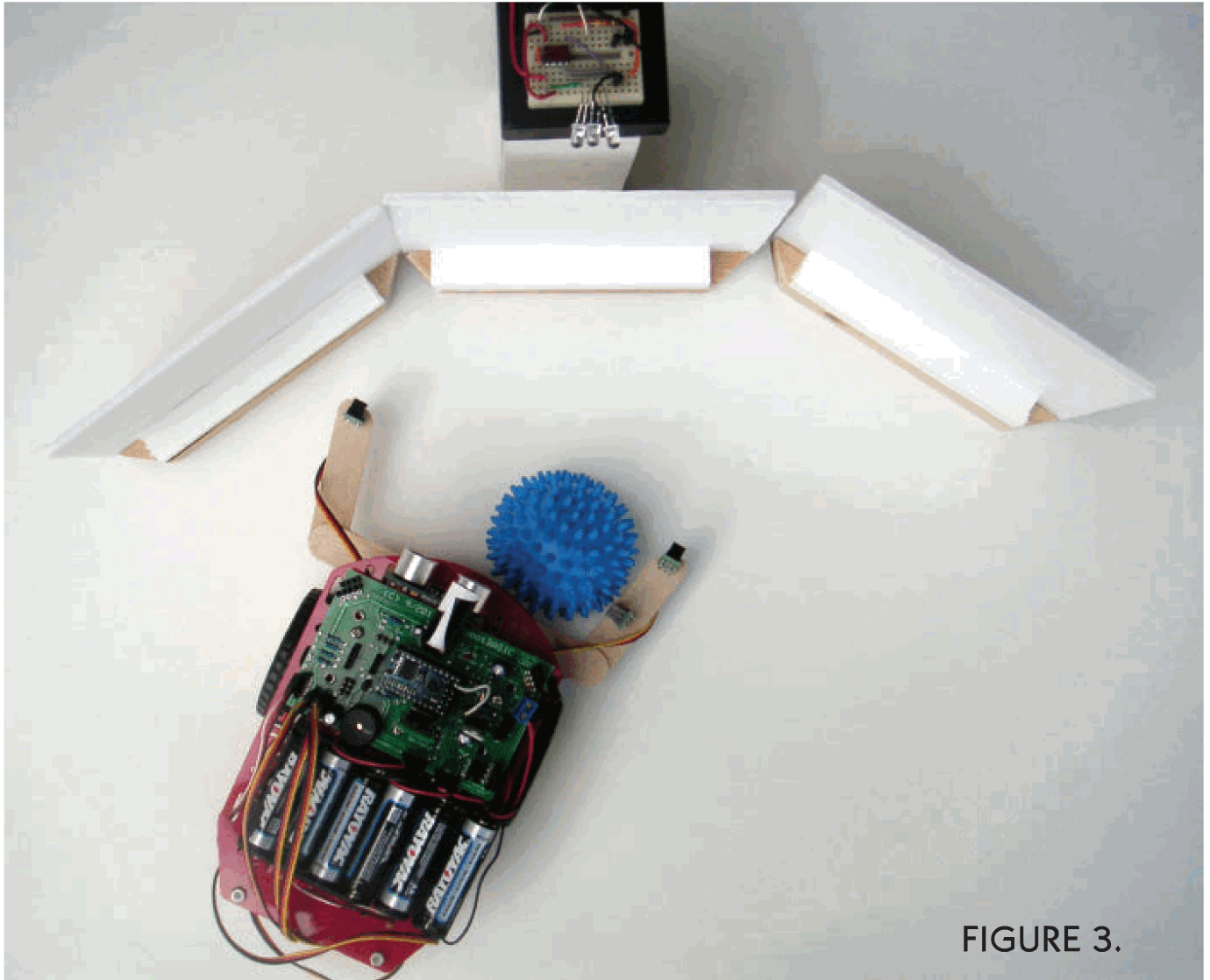


FIGURE 3.

Figure 3 shows the robot approaching the target goal where the ball is to be deposited. Notice the IR beacon mounted above and behind the goal. Complete information on how the RobotBASIC beacon system works can be found in the RROS manual which can be downloaded from www.RobotBASIC.com.

The ball we used has a nipped surface which prevents it from rolling

away from the robot too quickly. This made development of the algorithm much easier. The soccer goal is made from foamboard panels; we often use this material to create walls and obstacles for our projects.

Implementing the Algorithm

We used RobotBASIC to

implement the algorithm. Its easy-to-read syntax makes it simple to translate programs for use with other controllers. **Figure 4** shows how quickly the algorithm can be implemented if it is assumed that all the details are handled by separate subroutines.

The first part of **Figure 4** uses library routines to perform the initialization necessary for the RROS to control the Magician chassis and our chosen sensors. There is also a command to set an inverse mask. This is needed because we attached the trap sensors to the RROS line sensor inputs.

The line sensors report true when they see a black line (no reflection); we need to invert this logic with the trap sensors so they will report true when a reflection (from a ball or other object) is detected.

FIGURE 4.

```
#include "RROScommands.bas"
#include "InitializationRoutines.bas"
gosub InitRROScommands
gosub InitMagicianChassis
rCommand(SetSenseInvMask,7)

main:
  gosub GetObject // acquire the ball
  gosub MoveToBeacon // move it to the goal
end
```


Inside the main program, the subroutine **GetObject** is used to find a ball. Once a ball is found, the subroutine **MoveToBeacon** will move the ball to the goal. **Figure 5** shows how the ball is obtained. The outer *repeat loop* will continue looking until a ball is found.

Inside the loop, the speed is set to 25% of maximum to better locate objects. Also, the turn style is set to 100 so that the robot will rotate around its center when turning.

An inner *repeat loop* uses a *for loop* to rotate the robot left for a short time as it looks for a nearby object. You may need to adjust the *for loop* and the range distance to work for your particular situation.

If an object is not found, the robot turns a little and moves forward. This serves as a random movement to help the robot roam around until it “sees” an object to examine.

When the inner *repeat loop* terminates, it indicates an object has been found and a *for loop* moves the robot forward (for a short time) to see if it can find the object detected by the ranging sensor. The object is assumed to be found if any of the trap sensors detect it.

Identifying the Ball

Once an object is found, the program tries to identify it. Another *for loop* lets the robot briefly try to get the object into the trap. If it fits inside the trap, we will assume it is a ball.

This is a very simplistic way to distinguish the ball from other objects, but it serves to demonstrate the principle. A more robust algorithm could use a color sensor or other alternative approaches to help ensure that the desired object was found.

While trying to get the ball into the trap, the robot will attempt to adjust for situations it might encounter. If both of the front trap sensors are triggered, a large object (such as a wall or the goal) will be assumed, and the robot will back away and continue looking for the ball. If either of the front trap sensors are triggered, the robot will turn left

or right to try to center the object so it can fit into the trap as the robot moves forward.

There is an important aspect to this routine. The robot always uses *for loops* to limit the amount of time it tries to perform any given task. If the task being performed is not accomplished in a timely manner, the robot restarts the entire routine. While this may prevent the robot from occasionally obtaining a ball it has found, it also prevents the robot from getting stuck in a loop where the robot’s actions make no sense at all.

FIGURE 5.

```
GetObject:
  repeat
    rCommand(SetSpeed,25)
    rCommand(SetTurnStyle,100)
    // look for some object that is close by
    repeat
      found = 0
      for i=1 to 100
        rTurn -1
        if rRange()<40
          found=1
          break
        endif
      next
      if not found
        // move around a little to help find an object
        rTurn -20
        if rRange()>20 then rForward 20
      endif
    until found

    // some object now in view - move to it
    found = 0
    for i=1 to 100
      rForward 1
      if rSense()
        found=1
        break
      endif
    next

    if found
      // object found, now see if it might be a ball
      found = 0
      // try for a short time to get it in the hand
      for i=1 to 50
        if rSense()=1 // something is in the hand
          found = 1
          break
        endif
        if rSense()=6 // ignore big objects
          rForward -40
          rTurn -30
        endif
        while rSense()=2 // right trap sensor
          rTurn 1
        wend
        while rSense()=4 // left trap sensor
          rTurn -1
        wend
        rForward 1
      next
    endif
  until found // something is now in the hand
  return
```

Moving to the Goal

Once the ball is in the trap, the subroutine in **Figure 6** will find a beacon and move the robot toward it so that the ball can be deposited at the goal. At the beginning of this routine, the speed is increased slightly to help the robot keep the ball in the trap as it turns to look for the beacon. The turn style is also changed to zero, which simply causes the robot to rotate around one wheel rather than around its center.

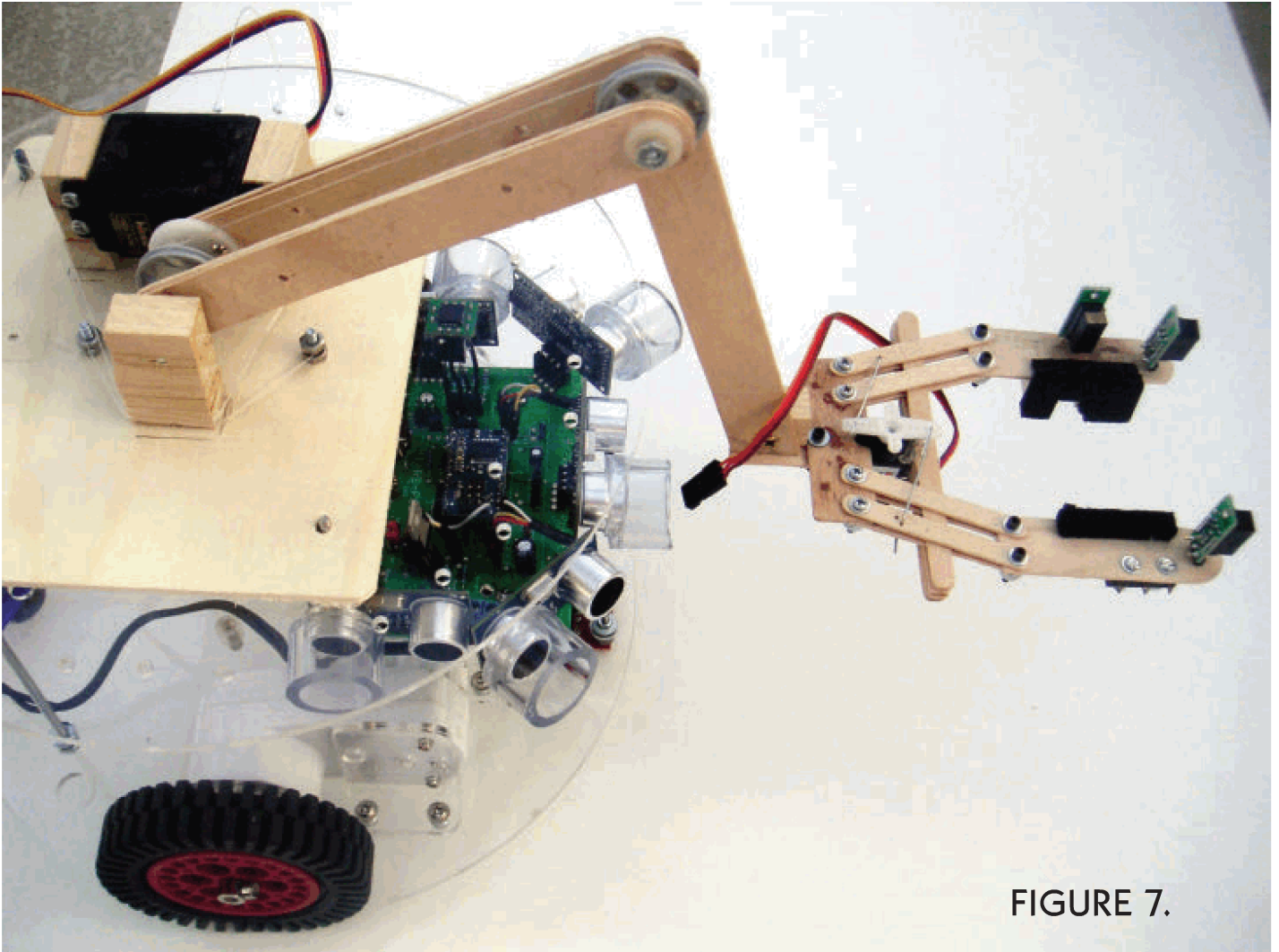


FIGURE 7.

FIGURE 6.

```
MoveToBeacon:
  // first face the beacon
  rCommand(SetSpeed,40)
  rCommand(SetTurnStyle,0)
  while not rBeacon(0)
    rTurn -1 // rotate to left
  wend
  // then move to it
  rTurn 10
  rCommand(SetTurnStyle,0)
  while (rSense()&1) and not (rSense()&6)
    rForward 1
    if rBeacon(0)
      rTurn 1
    else
      rTurn -1
    endif
  wend
  return
```

This motion helps keep the ball in the trap. The robot will turn left in this manner as it looks for the beacon located at the goal.

When the beacon is found, the robot turns back to the right slightly to ensure that it is aimed at the

beacon or a little to the right of it. It then moves forward, turning to the left when the beacon is not seen, and to the right when it is. This action effectively lets the robot follow the beacon's signal. It does so until one or more of the front trap sensors indicate

that the goal has been reached.

Conclusion

We have provided an easy-to-understand example of how a robot can locate and identify a ball within its environment, and then move it to a goal. A more robust algorithm for obtaining the ball could certainly be implemented — especially if additional sensors are utilized.

The hardware requirements for this project are minimal, so you might find you can experiment with these principles using a different robot. Watching the robot seek out the ball, maneuver to trap it, and move it to the goal is definitely worth the effort.

Next month, we will build on the ideas discussed here and show how a robot (see **Figure 7**) can find small objects, then retrieve them with an easy-to-build arm. **SV**

Model 324

WORLD'S MOST VERSATILE CIRCUIT BOARD HOLDERS

Our line of Circuit Board Holders add versatility & precision to your DIY electronics project. Solder, assemble & organize with ease.

MONTHLY CONTEST
Visit us on Facebook® to post a photo of your creative PanaVise project for a chance to win a PanaVise prize package.

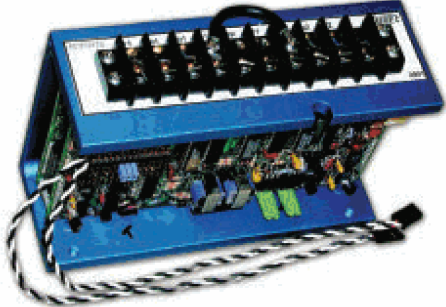
PANAVISE®
Innovative Holding Solutions

Model 201

VISIT US ON  

7540 Colbert Drive • Reno • Nevada 89511
1 (800) 759-7535 | www.PanaVise.com

STEER WINNING ROBOTS WITHOUT SERVOS!



Perform proportional speed, direction, and steering with only two Radio/Control channels for vehicles using two separate brush-type electric motors mounted right and left with our **mixing RDFR dual speed control**. Used in many successful competitive robots. Single joystick operation: up goes straight ahead, down is reverse. Pure right or left twirls vehicle as motors turn opposite directions. In between stick positions completely proportional. Plugs in like a servo to your Futaba, JR, Hitec, or similar radio. Compatible with gyro steering stabilization. Various volt and amp sizes available. The RDFR47E 55V 75A per motor unit pictured above.
www.vantec.com

VANTEC **Order at (800) 882-6832**


THE ORIGINAL SINCE 1991
PCB-Pool®
Beta LAYOUT

FREE Stencil
with every prototype order

Embedded RFID
authenticate, track & protect your product

www.magic-pcb.com

Call Tyler: 1 707 447 7744
sales@pcb-pool.com

AndyMark
Inspiring Mobility

Specializing in Unique Wheels, Gearboxes, Aluminum Sprockets and Drive Bases

c-Rio-Ready Full Chassis Kit

8" Plastic Omni Wheel

6" HD Mecanum Wheel Set

Aluminum Sprockets

AndyMark, Inc.
sales@andymark.com **Toll Free: 877-868-4770**

www.andymark.com

The *SERVO* Webstore

Attention Subscribers ask about your discount on prices marked with an *

CD-ROM SPECIALS

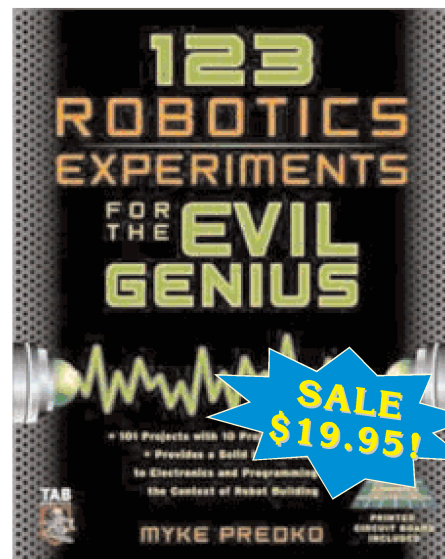


Free Shipping!

9 CD-ROMs & Hat Special

Only \$ 189.95
or \$24.95 each.

02.2013 FEATURED



123 ROBOTICS EXPERIMENTS FOR THE EVIL GENIUS

SALE \$19.95!

101 Projects with 10-Page
Provides a Solid
to Electronics and Programming
the Context of Robot Building

MYKE PREDKO

ROBOTICS

Robot Builder's Bonanza, Fourth Edition

by Gordon McComb

Robot Builder's Bonanza, Fourth Edition includes step-by-step plans for a number of motorized platforms. The book is described as a compendium of robotics topics, containing more than 100 projects, including 10 robot designs new to the fourth edition. These modular robots are low cost, and are made to be reproduced by readers with no training in mechanical construction.

\$29.95*

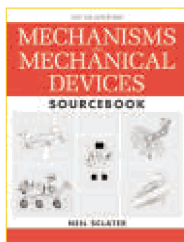


Mechanisms and Mechanical Devices Sourcebook 5th Edition

by Neil Sclater

Fully revised throughout, this abundantly illustrated reference describes proven mechanisms and mechanical devices. Each illustration represents a design concept that can easily be recycled for use in new or modified mechanical, electromechanical, or mechatronic products. Tutorials on the basics of mechanisms and motion control systems introduce you to those subjects or act as a refresher.

Reg \$89.95 Sale Price \$79.95



Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists

by Dustyn Roberts

In *Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists*, you'll learn how to successfully build moving mechanisms through non-technical explanations, examples, and do-it-yourself projects — from kinetic art installations to creative toys to energy-harvesting devices. Photographs, illustrations, screenshots, and images of 3D models are included for each project.

\$29.95*



Build Your Own Humanoid Robots

by Karl Williams

GREAT 'DROIDS, INDEED!

This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than \$300. Together, they form the essential ingredients for making your own humanoid robot.

\$24.95*



Robot Programmer's Bonanza

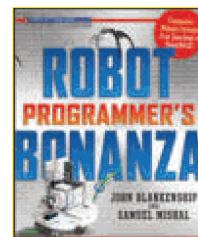
by

John Blankenship,
Samuel Mishal

The first hands-on programming guide for today's robot hobbyist!

Get ready to reach into your programming toolbox and control a robot like never before! *Robot Programmer's Bonanza* is the one-stop guide for everyone from robot novices to advanced hobbyists who are ready to go beyond just building robots and start programming them to perform useful tasks.

\$29.95



Robotics Demystified

by Edwin Wise

YOU DON'T NEED ARTIFICIAL INTELLIGENCE TO LEARN ROBOTICS! Now anyone with an interest in robotics can gain a deeper understanding — without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots!



\$19.95

We accept VISA, MC, AMEX,
and DISCOVER
Prices do not include shipping and
may be subject to change.

To order call 1-800-783-4624

SERVO Magazine Bundles



**Save \$10.00
Only \$57.95!**

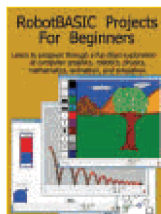
Now you can get one year's worth of all your favorite articles from *SERVO Magazine* in a convenient bundle of print copies. Available for years 04, 05, 06, 07, 08, 09, 10, 11 and 2012.

RobotBASIC Projects For Beginners

by John Blankenship,
Samuel Mishal

If you want to learn how to program, this is the book for you. Most texts on programming offer dry, boring examples that are difficult to follow. In this book, a wide variety of interesting and relevant subjects are explored using a problem-solving methodology that develops logical thinking skills while making learning fun. RobotBASIC is an easy-to-use computer language available for any Windows-based PC and is used throughout the text.

Price \$14.95

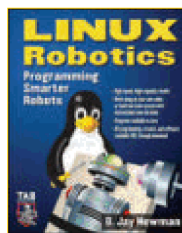


Linux Robotics

by D. Jay Newman

If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for “Zeppo,” a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need.

\$38.95

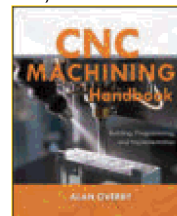


CNC Machining Handbook: Building, Programming, and Implementation

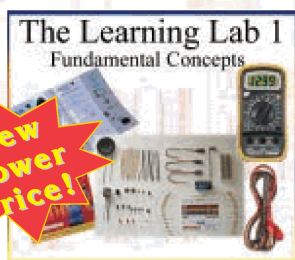
by Alan Overby

The *CNC Machining Handbook* describes the steps involved in building a CNC machine and successfully implementing it in a real world application. Helpful photos and illustrations are featured throughout. Whether you're a student, hobbyist, or business owner looking to move from a manual manufacturing process to the accuracy and repeatability of what CNC has to offer, you'll benefit from the in-depth information in this comprehensive resource.

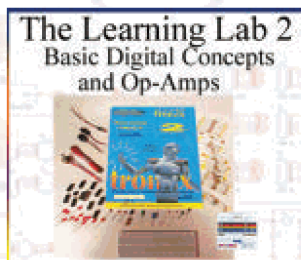
\$34.95



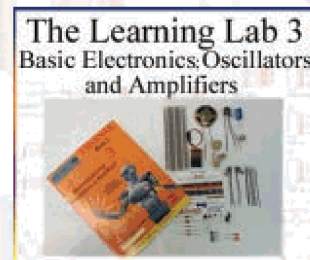
FOR BEGINNER BOT BUILDERS



\$59.95



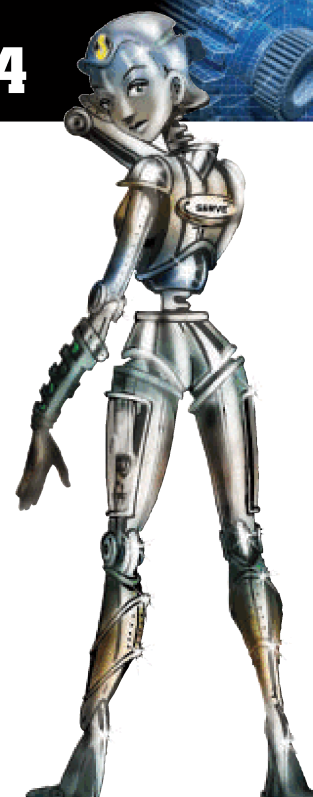
\$49.95



\$39.95

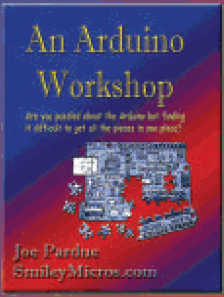
The labs in this series — from GSS Tech Ed — show simple and interesting experiments and lessons, all done on a solderless circuit board. As you do each experiment, you learn how basic components work in a circuit, and continue to build your arsenal of knowledge with each successive experiment.

For more info and a promotional video, please visit our webstore.



Or order online
www.servomagazine.com

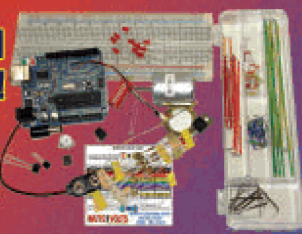
SPECIAL OFFERS



Puzzled by the Arduino?

Based on *Nuts & Volts*
Smiley's Workshop,
 this set gives you all
 the pieces you need!

Book and Kit Combo \$124.95



Forbidden LEGO
 by Ulrik Pilegaard / Mike Dooley

Forbidden LEGO introduces you to the type of free-style building that LEGO's master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse.

Reg \$24.95 Sale Price \$19.95



Beginner's Guide Complete Combo!



Combo Price \$229.95

"THE PHANTOM DRAW"

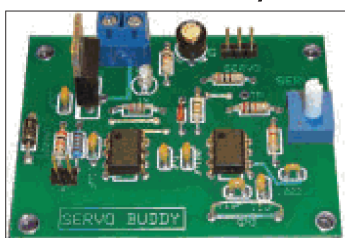
The KILL A WATT
 meter is the best way to help you determine your actual energy draw in ON and OFF home appliances.

Only \$29.95

To order call
 1 800 783-4624 or online www.servomagazine.com

PROJECTS

The SERVO Buddy Kit



An inexpensive circuit you can build to control a servo without a microcontroller.



For more information,
 please check out the
May 2008 issue
 or go to the
SERVO webstore.

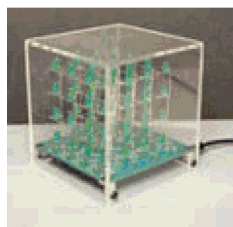
Includes an article reprint.

Subscriber's Price **\$39.55**

Non-Subscriber's Price **\$43.95**

3D LED Cube Kit

From the
 article "Build
 the 3D LED
 Matrix Cube"
 as seen in the
August 2011
 issue of
Nuts & Volts Magazine.

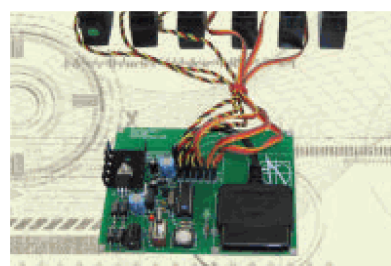


This kit shows you how to build a really cool 3D cube with a 4 x 4 x 4 monochromatic LED matrix which has a total of 64 LEDs. The preprogrammed microcontroller that includes 29 patterns that will automatically play with a runtime of approximately 6-1/2 minutes. Colors available: Green, Red, Yellow & Blue. Jig and plastic cases also available.

Subscriber's Price **\$57.95**

Non-Subscriber's Price **\$59.95**

PS2 Servomotor Controller Kit



This kit accompanied with your own PlayStation controller will allow you to control up to six servomotors.

Includes all components and instruction manual.

For more information, please see the February 2011 edition of *SERVO Magazine*. Assembled units available!

Subscriber's Price
\$79.95

Non-Subscriber's Price
\$84.95



Twin Tweaks

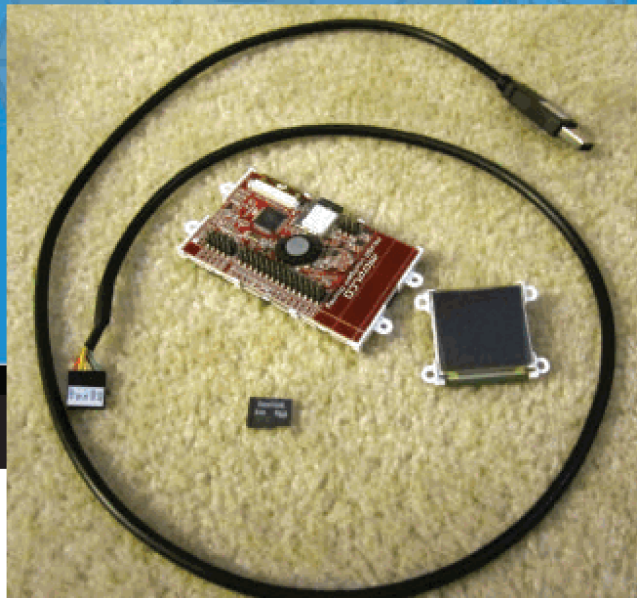
BRYCE WOOLLEY & EVAN WOOLLEY



www.servomagazine.com/index.php?/magazine/article/march2013_TwinTweaks

THIS MONTH:

Robots in 4D



SMART OLED DISPLAY MODULES FROM 4D SYSTEMS.

Most of the time when we get super cool stuff to tinker with for this column, the connection to robotics is obvious. The connection is most apparent, of course, when the super cool stuff we're working with is an actual robot, but even when the star of our column isn't a full-on robot, its place in the field is usually quite clear.

We've worked with individual sensors like the position controller from Parallax (check out the July 2010 issue for the full story), and even mechanical modifications like omni wheels and an automotive steering mod for the VEX Robotics Design System (September 2007 issue). This time, however, we had the challenge of implementing something in a robot that — at first glance — looked a little more stand-alone than a simple sensor.

This month, we had the opportunity to work with the smart OLED and LCD display modules from 4D Systems. 4D Systems is an Australian based company that — according to their slogan — seeks to turn technology into art.

Do art and robots mix? We're about to find out.

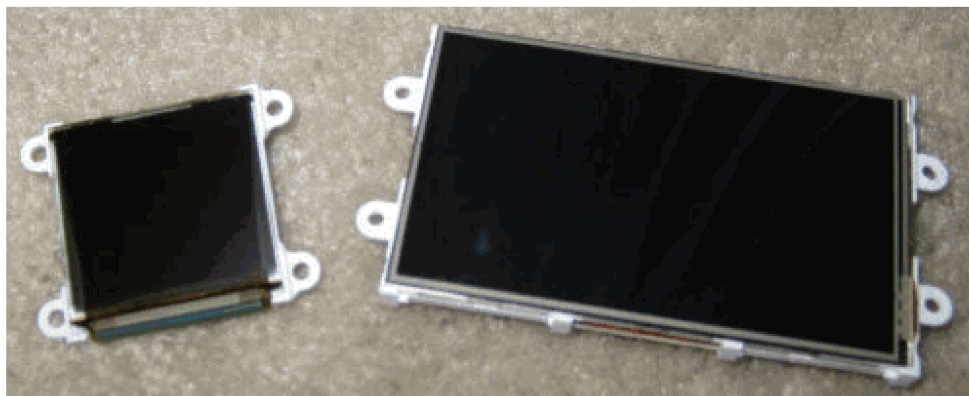
This is 4D

4D Systems offers a variety of OLED and LCD smart display modules to add pizzazz to a wide

range of projects. The modules come in screen sizes ranging from 0.96" to 4.3" and they are definitely more than just a screen. They come attached to a circuit board with an embedded graphics processor.

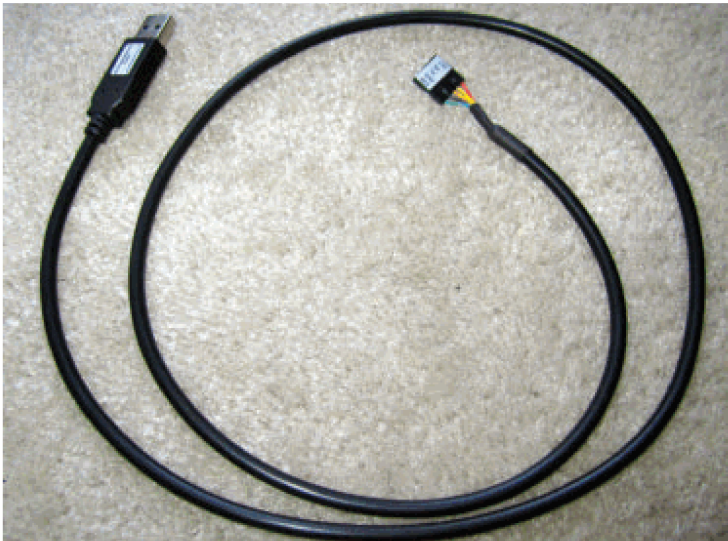
The graphics processor comes in two flavors: the whimsically named GOLDELOX and the fitting PICASO. The modules include a variety of peripherals that make the smart displays great candidates for pretty much any project.

There is a port for a micro-SD card to add up to 2 GB



THE μOLED-128-G2 MODULE AND THE μLCD-32WPTU MODULE.

THE 4D PROGRAMMING CABLE.



of memory to the device; there are general-purpose ins/outs that could be connected to digital sensors; and there are ports for A/D conversion in case analog sensors are your cup of tea. There's even the capability for sound generation, just in case being a visually gifted device was not enough for your Renaissance robot.

We received two units from 4D Systems: the μ OLED-128-G2 module with a 1.5" display, and the μ LCD-32WPTU module with a 3.2" display. LCD screens have been a fairly frequent addition to many robots because they're a great way to send messages to the user, debug the bot, or otherwise monitor what's happening. These, however, were not your average LCD modules.

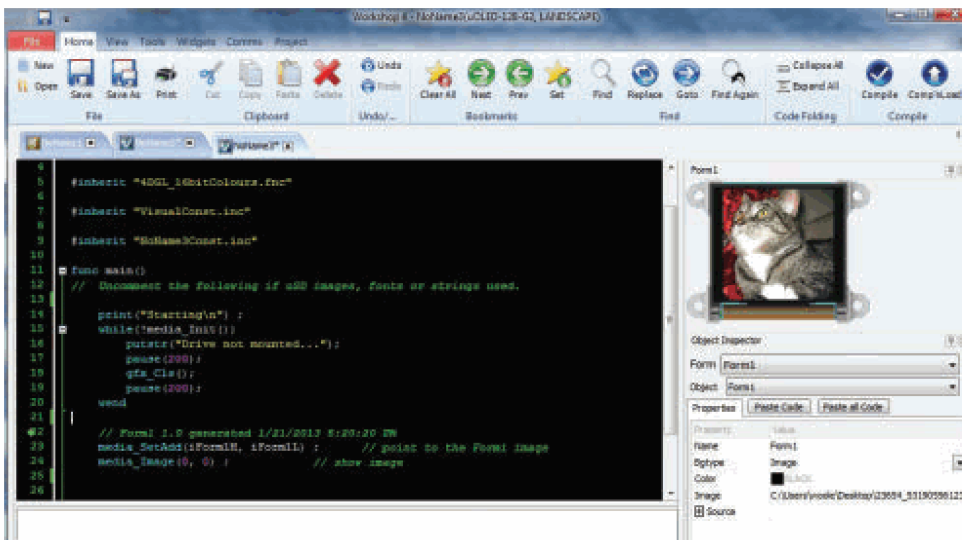
In addition to the features mentioned above, the LCD modules from 4D have resistive touch screens. Thus, while LCD screens have appeared in some of our projects

as passive displays, these modules could be used as interactive user interfaces. The other, smaller module was an OLED screen. Every tinkerer worth their salt has come across and used LEDs in their projects. The classic LED looks like a tiny cylindrical bulb with a rounded top that emits bright light of a certain color. The 4D smart display, however, was not just any LED — it was an OLED, or Organic Light Emitting Diode.

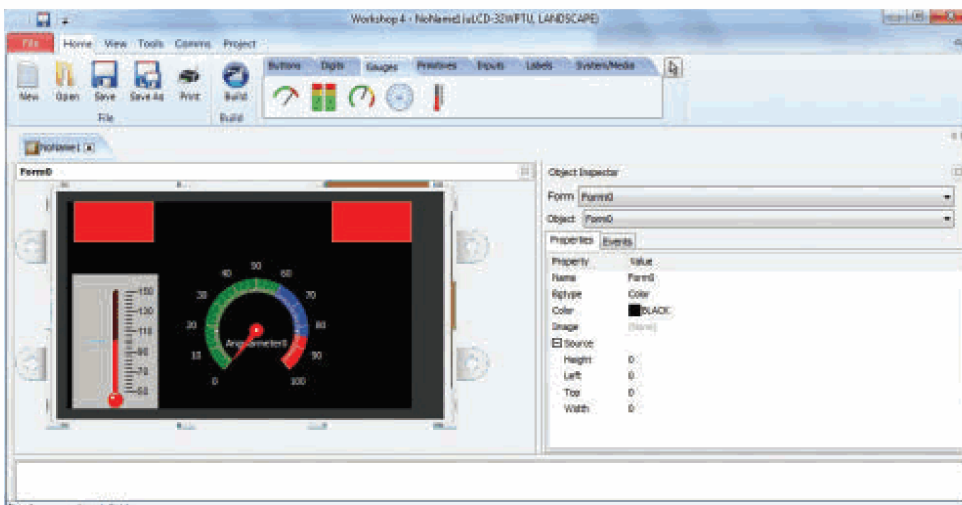
As its name suggests, the OLED is made out of an organic compound. The most noticeable difference between the OLED and the traditional LED is that the OLED comes in a thin sheet and looks more like a thin layer of plastic than the familiar bulb.

The OLED also boasts some other favorable characteristics, like no need for a backlight and better contrast than LCD screens.

All of these features sound great, but they also make it seem like the promise of cost-effective modules might be out of reach. Particularly with the fairly recent emergence of OLED technology, we figured the idea of cost-effectiveness was more relative than anything else. The wide screen LCD module, however, can be acquired from the 4D website for \$89, and the OLED module is only \$55. With a price like that, the modules had our attention, as well as our curiosity.



PROGRAMMING THE OLED MODULE WITH VISI.



PROGRAMMING THE LCD MODULE WITH VISI GENIE.

RECOMMENDED
WEBSITE

4D Systems
www.4dsystems.com.au

Visi-ualizing Success

A piece of hardware can have the best specs in the world, but without the right programming it can be little more than a fancy paperweight. Users have a few options for programming the 4D display modules, and all of them are within the universe of the Workshop 4 IDE. The Workshop 4 IDE is a custom Integrated Development Environment that caters to users of all programming skill levels.

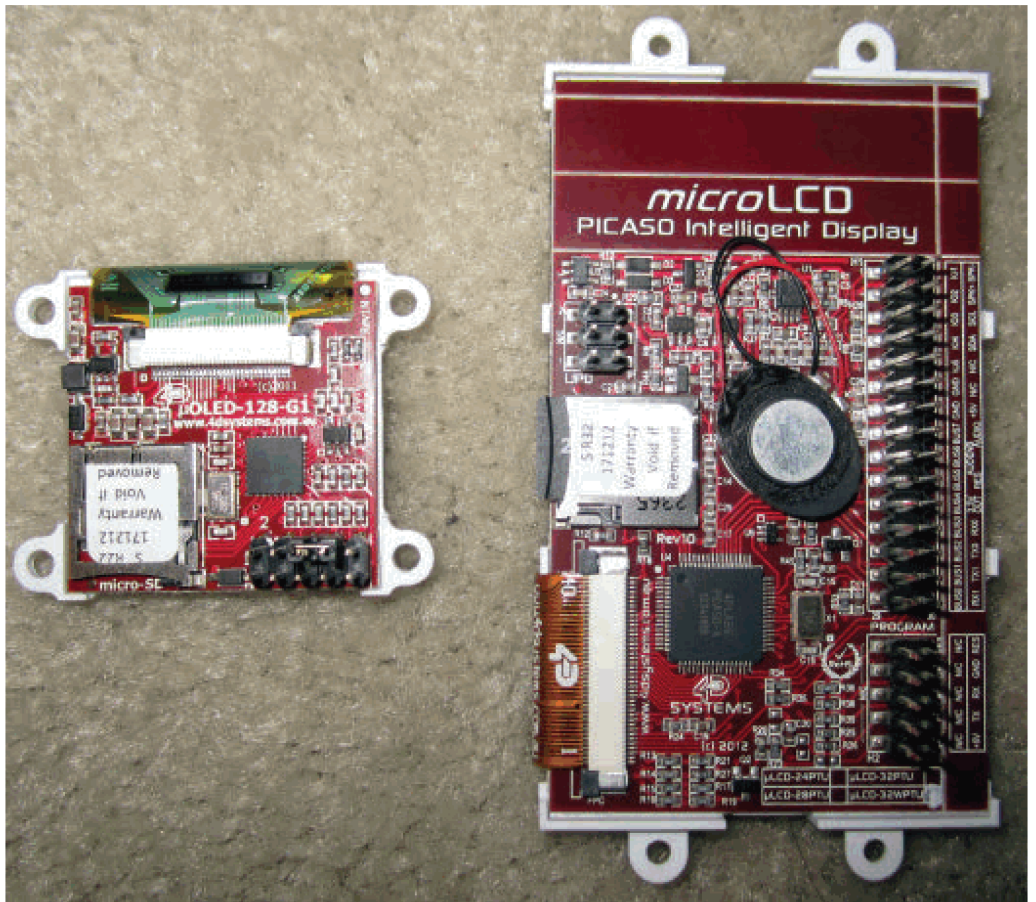
The traditional development environment is the Designer, where users write textual code. The code in this case is 4DGL — a custom graphics language that draws familiar elements from C and Java. Even for a purely textual environment, the Designer has great visual flair. Different parts of the syntax come in different bright colors that stand in stark contrast to the black background. The 4D Systems website features numerous guides to introduce users to the syntax of the Designer and an entire code library of sample programs. Even so, coding in pure text can be daunting to the uninitiated.

For that reason, the Workshop 4 also includes Visi — a hybrid of the text-based Designer and a more graphical interface that allows users to create the display they want by dragging and dropping elements onto a template that looks like the target smart display. The elements that can be dragged and dropped include simple shapes, interactive buttons for the touch screen, indicator lights, and gauges for everything from rpm to temperature. Once you've created the display you want, the touch of a button will paste all the code to make it happen right into the Designer side of the IDE.

For the completely text averse, the Workshop 4 also includes the Visi-Genie which is a completely graphical way to construct the module display. The Visi-Genie is much like Visi, except it does not show a window with the code in text form.

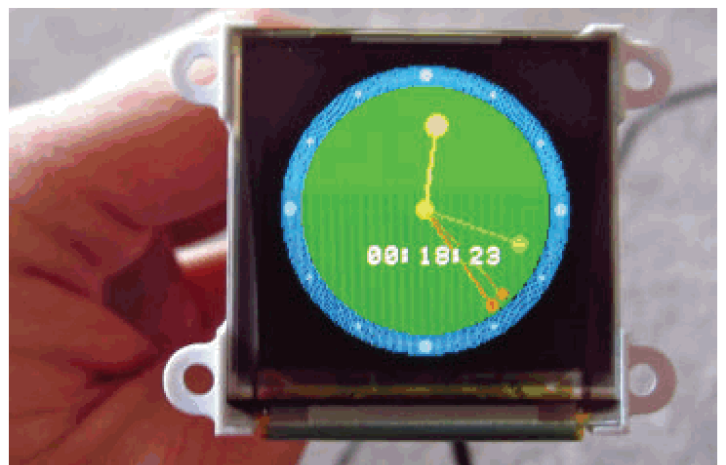
To get the code from the computer to the module,

PUTTING THE SMART IN SMART MODULE.



4D Systems offers a custom programming cable. We were ecstatic to see that the cable smartly used a USB connection to plug into the computer. The other end of the cable is a five-pin header that connects to the display module. The cable handily sources 5V to power the module when programming.

We opted to use Visi for our experimenting purposes to get a sense of both the textual programming and the usefulness of the drag-and-drop graphical interface.



THE MODULE SPRINGS TO LIFE!



TESTING THE MODULE WITH A PICTURE.

The Robot With a Face

After becoming thoroughly acquainted with the smart display modules, it was time to figure out how to implement them with a robot. A lot of the more intuitive ways to implement the displays would be as just that — passive displays. That, however, didn't seem to take full advantage of the smart nature of these devices.

We thought the perfect way to incorporate the smart displays into a robot would be a whimsical way of responding to sensor input. Simple displays and LEDs are often used for such a purpose, often as a debugging tool — when a button gets pressed or when an IR sensor reads high or low, an LED can come on to indicate that the sensor is working. Something like that might be one useful implementation of the smart displays. Instead of adding an LED for every sensor your robot uses, you can simply add

more indicators to the display. Dragging and dropping additional indicators would be a lot easier than wiring up and powering another LED for every sensor you want to add.

We thought a more whimsical take on this classic implementation would be to put a face onto the display that would react to sensor input. Our initial vision was a face that used both screens — the smaller OLED module could show a cycloptic eye, and the widescreen LCD screen would show an expressive mouth. Our goal was to have the bot change expressions in response to sensor input.

When a touch sensor was pressed, the eye could glance over to the sensor and the mouth could express a grimace (or joy, if you prefer for your robot to appreciate tactile interaction). We also thought that an infrared sensor could provide some amusing possibilities — perhaps a robot accustomed to dark

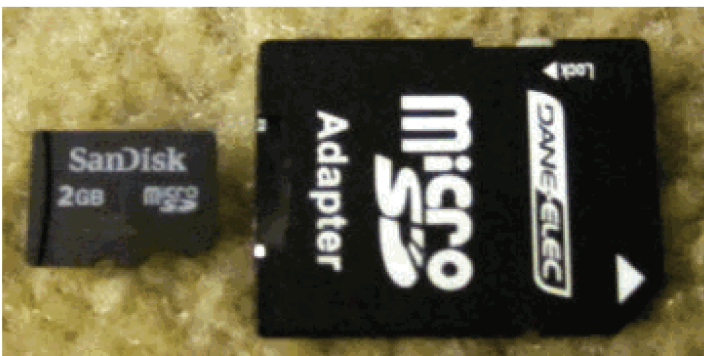
surfaces could squint in discomfort when coming across a bright one.

Implementing the screens on a mobile robot, however, would take a little bit of problem solving. The difficulty in implementing the OLED and LCD screens on the robot had its origin in power requirements. Some elements are not overly power hungry, like sensors. Sensors will usually take about five volts or less. In fact, the OLED and LCD screens themselves operate off of five volts. Both modules also come equipped with a host of digital ins/outs just begging for some sensors.

It looked like things were shaping up to be a five volt party, but something was missing. Sensors are a critical part of a robot, but in our minds an ideal robot should be mobile. The OLED and LCD screens, however, do not have the capability to source the power necessary to drive motors (nor do they have motor drivers).

At first blush, this didn't seem like much of an issue. Since the smart displays don't have motor drivers anyway, chances are that you'll want to use some sort of brain meant for a robot. That brain will likely be accompanied by a battery with the juice to quench the thirst of power hungry motors. The modules are smart, but two brains are better than one, right?

Two brains are fine and dandy, but then the project would become something more like two robots Frankensteined together. What we wanted to achieve was a harmonious and symbiotic whole — like Eddie Brock and the Venom Symbiote (well, more harmonious than that). And just like a good symbiote, we wanted both organisms to use the same power source and communicate with each other.



ADAPTATION.

It Will Become You

Now that we had the plan, we needed to get the two halves of our robot symbiote ready. We opted to start with the smart display modules. The software and all of the supporting documentation discussed above can be downloaded for free from the 4D website. For our initial testing we chose to use the OLED module, largely because we were enticed by the novelty of the relatively new technology and we wanted to see what kind of image quality the little screen could really produce.

For our first test, we used the classic “Hello World” program. We figured a simple program displaying only text wouldn’t require us to fiddle with the SD card just yet. One handy thing about starting new programs in the 4D Workshop is that you’re given a little more to start with than just a complete tabula rasa. A library of functions for the graphics processor and a main program loop are already included. All it took was a few more lines of code to instruct the screen to display that immortal phrase, and we were ready for our first test.

Programming robots can often be a tedious process, with syntax errors and COM port conundrums delaying the gratification of seeing something actually work until long into the process. This was not the case with the smart display OLED module, though, because just as it was plugged in it sprang to life with what must be a default program. The default program in this case was not some unremarkable color test or some such banality, but rather an impressive display of the capabilities of the screen.

First, there was the 4D Systems logo and an explosion of color akin to a Fourth of July firework, and then a brightly colored clock started keeping time with sweeping second and minute hands. The swirling motion and bright colors of the clock were a cool introduction to the capabilities of the screen, but what we really wanted to see was the output from our own program.

With the screen connected, we compiled and loaded the program. The window at the bottom of the screen assured that no errors were found and that the program loaded successfully. When we looked to the screen, we were happy to see the immortal “Hello World” emblazoned on a black background in defiant green type.

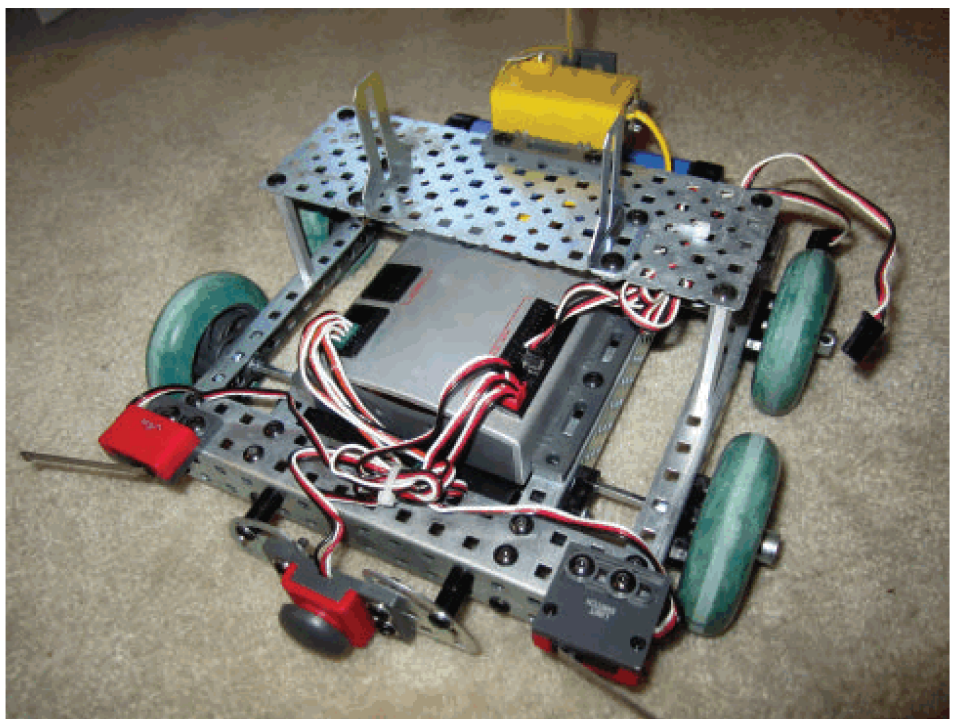
With the success of our first test behind us, we wanted to see what kind of image quality the smart display could muster. Using Visi, we were easily able to pick an image from our



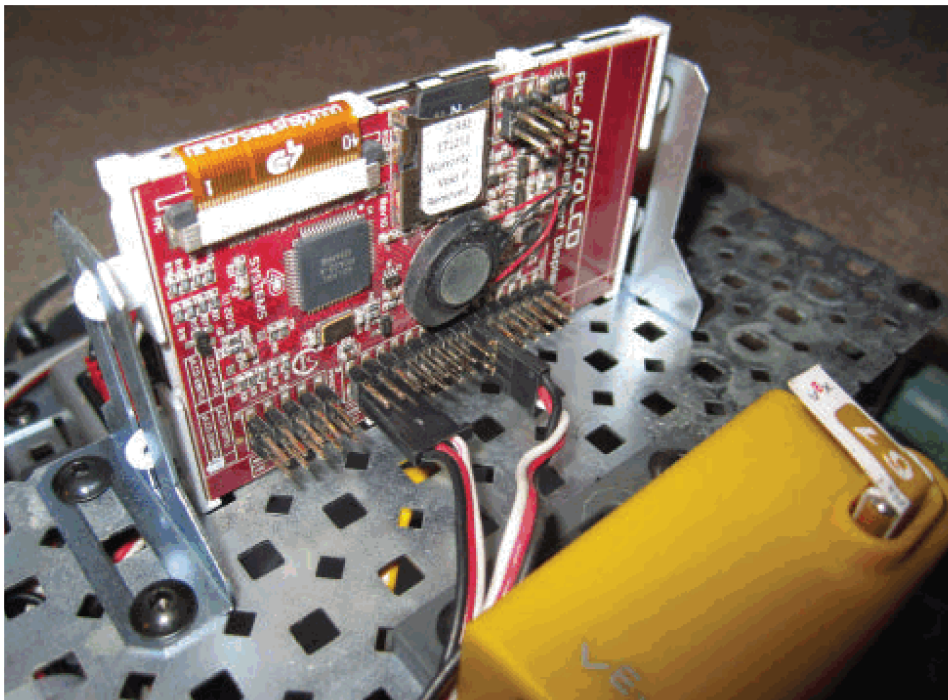
THE PROGRAMMING SETUP.

computer to load onto the screen — a shot of our cat Lida posing glamorously in front of a pillow. Getting the image onto the screen, however, was not quite as easy as just picking a photo from your computer.

The smart modules have enough memory on board to



A ROBOTIC BASE THAT NEEDS A FACE.



WIRING UP THE LCD MODULE.

handle simple programs, but images and videos take a lot more memory. To accommodate all of the images and video you could ever want to see on a tiny screen, the modules smartly come with the capability of attaching

a micro-SD card for additional memory.

One of the exciting things about the option to use an SD card is that the screen can be used to show off sophisticated graphics. Using a large SD card, you could even play a movie. Add in a speaker for sound (which the module has the pins for), and you've given a whole new spin to personal entertainment systems.

The trick about loading images onto the smart display module is realizing that you're really loading them onto the SD card. Thus, the images and other graphics that you want displayed on the screen first need to be loaded onto the micro-SD card instead of drawn directly from your computer.

A micro-SD card truly lives up to its name, so be sure that you have an adapter card for it so that it doesn't get gobbled up by your computer like a Hungry Hungry Hippo.

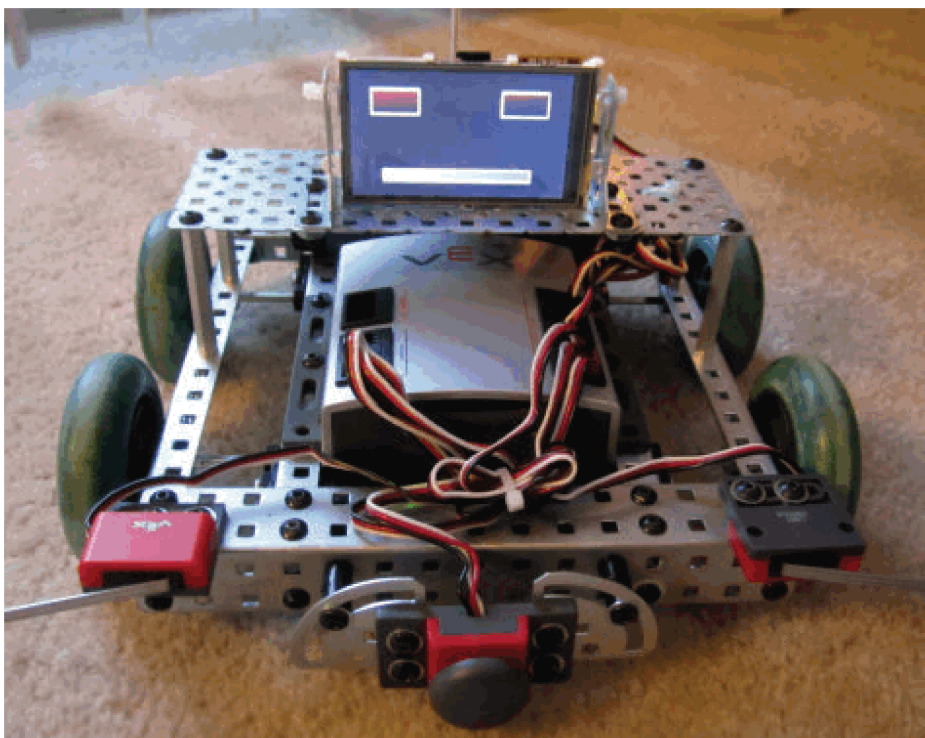
Once the image has been loaded onto the SD card, you can set it to the background of the screen in Visi. The image will appear in the interface, and you have the ability to crop and position it.

When it all looked good, we pasted the code into the program loop. There were no errors upon compiling the program, and we were thrilled to see a little picture of our cat appear on the screen. The brightness and resolution of the screen were impressive, and we were excited to see what kind of face we could add to our robot.

To add a face to our robot, though, we needed a robot. We thought that a robot built from the VEX Robotics Design System would be the perfect fit. We had touch and IR sensors that we could use with it; the VEX brain should be easy enough to interface with the smart modules; and the ability to program the whole thing in Easy C was just icing on the cake.

We put together a simple four wheel direct drive base from the VEX kit. For starters, we added some touch sensors to the front of the bot for some tactile interaction.

With the bot ready, all we needed was a face.



I'M SORRY DAVE, MY EMOTIONAL RANGE IS LIMITED RIGHT NOW.

Face/On

The wide screen LCD module seemed the most capable of handling an entire face at once. After we got one module sorted out, we would move onto our dream of an expressive Cyclops using both modules. Using the Visi Genie, we crafted a simple face out of a few rectangles. Two red rectangles for eyes and a textured rectangle for a mouth created a rudimentary face with an emotional range slightly more limited than that of the Master Control Program.

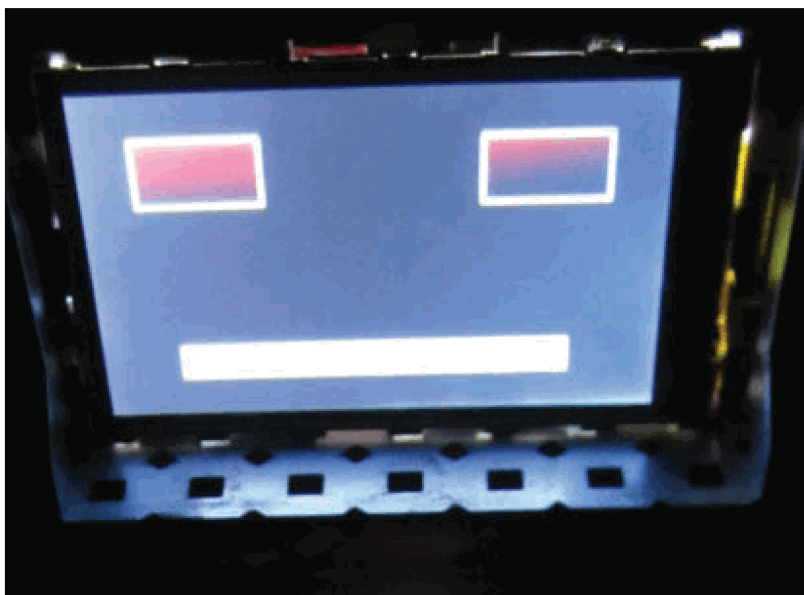
Before we ventured too far down the programming rabbit hole, we thought it would be useful to sort out the power requirements. The display module requires only 5V, which is the same amount you need to power most sensors. Thus, for the purposes of powering the display module, it could be treated like any other sensor and hooked up to a 5V source. On the VEX brain, there is a plethora of open ports for sensors, so we picked one and connected the other end to the 5V and ground pins on the module. We also hooked up the Tx and Rx ports of the brain and the smart module to each other to enable communication between the parts of the robotic symbiote.

The Future is on Display

Up to this point, we had cleared one of the major hurdles to integrating the smart display into a robot — we had successfully sourced power from the VEX robot battery to the display. The next steps will be to add some emotional range to the bot by having it change expressions based on sensor input. This would take some tricky wiring and some even trickier programming. All that cool factor stuff was simply too much to fit into one article (though we promise not to pull a Hobbit and split it into three). Next time, we'll devise a way for the two halves of the symbiote to respond to the same sensor input; we'll add a more detailed face; and then see if we can get both smart modules to coexist with the VEX brain.

Even though we had just scratched the thin, flexible surface of the 4D Systems smart modules, we could already see the potential for a myriad of robotics applications. The folks at 4D also show a keen eye for the robotics potential of their OLED and LCD modules. Some upcoming new products are Arduino shields and starter kits incorporating the OLED and LCD modules.

With our first glimpse into the image quality and



SOMEBODY'S WATCHING ME...

programming with the smart modules, we feel safe to say that they'll be a good way to add pizzazz to any robot. **SV**



AP CIRCUITS
PCB Fabrication Since 1984

As low as...

\$9.95
each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com



SPECIAL THANKS TO

Atilla Aknar



Then and NOW

Controlling Robots in the Air and on the Ground

by Tom Carroll

Last October, I wrote about the many ways to control a robot and received some much appreciated constructive feedback on the article.

According to one reader, "the inclusion of quadcopter flight controllers and electronic speed controllers with higher power capacity for large robots would have been nice."

With that in mind, I will go a bit in-depth about some of the R/C systems required for larger robots and combat robots. I am also going to include flying multi-rotor R/C aircraft — a category I have skipped in past articles. I want to stick to standard R/C transmitters, receivers, and associated motor controllers that readers have favored in this follow-on article. Flight controllers are a special type of device I will also discuss.

Radio Control Helicopters Started the Multi-rotor Craze

In the November issue of *SERVO*, Michael Simpson started an excellent series on multi-rotor flyers and detailed some of the complex systems required to keep these quad and hex beasts flying. If you haven't done so, I suggest you look at these back issues as they have a lot of good information about flight controllers, electronic speed controllers, and radio gear that I won't be discussing in this article.

These multi-rotor flyers are proving to be the next level of robotics, and people have added cameras, sensors, and many other things such as high level microcontrollers for autonomous flight. Advanced flight controllers with GPS capabilities allow the flyer to head

back home autonomously when a mission is completed, or if the batteries are running low and a crash could be imminent.

Multi-Rotor Craft Can be Difficult to Fly

If you have flown an R/C helicopter that has a single main rotor and uses a tail rotor and collective pitch, you know just how hard it is for a beginner to keep things level and in control. My five-channel helicopter has managed to crash many times, and I've had to replace the stripped main rotor gear and the rotor and tail blades. My four-channel copter has crashed more than that.

Most lightweight helicopter structures are not very forgiving when they meet the ground at high speed.



FIGURE 1. Walkera CB 180.



FIGURE 3. New DJI Phantom UAV.



FIGURE 2. Blade 450 collective pitch helicopter.

Multi-rotor flyers do not use a tail rotor to steer (yaw), but must rely on driving each of the four, six, or more rotors at different speeds to tilt and rotate. These types of flying robots can also require up to six or seven R/C channels, depending on add-on equipment installed.

Figure 1 shows a Walkera CB 180Z that I've had for a while. I've placed ping-pong balls on the plastic rods to help prevent it from flipping over after a hard landing. This copter has eaten two sets of rotor blades and a main gear. The brushless drive motor used in these is quite powerful and can chew through the main gear like a Dremel tool when the rotor is stuck on the ground.

The Blade 450 shown in **Figure 2** has a main rotor with collective pitch and a tail rotor. It's quite a bit trickier to fly but is a lot of fun. Collective pitch allows you to vary the pitch of the main rotor's blades, which means you can flip the unit over in the air and fly it upside down.

A Flight Controller Makes Flying a Multi-Rotor Easier

Take a look at the new DJI Phantom UAV shown in **Figure 3**. There is no tail rotor to steer it. Unlike standard helicopters,

quadcopters and multi-rotor flyers use electrically-driven fixed-pitch props for all the lift force, and have no mechanical linkages to tail rotors or complex collective pitch and swash plate mechanisms.

So, how do you fly one of these neat quadcopters? This is where the flight controller — or flight stabilization controller — comes in. The result is a very maneuverable and agile aircraft.

Multi-Rotor Flight Stabilization Controller

There are numerous flight controllers that I've found in my research, and prices range from less than \$30 to many hundreds of dollars. Simpson's articles mentioned several good ones, including the DJI NAZA. Another great controller is the

MultiWii Pro shown in **Figure 4** with the attached MKT GPS module that has the 'go home' capability at a reasonable cost. The KK2 2.0 multi-rotor controller shown in **Figure 5** has an LCD panel that shows the configuration (and other properties). You can also see the typical interconnections from the receiver's servo connectors and the outputs to up to eight outrunner motors.

Parallax Enters the Market with the ELEV-8 Quadcopter

Parallax came out with the ELEV-8 shown in **Figure 6** last year, after fine-tuning the many mechanical aspects and sub-systems. Control of the vehicle's motion is achieved by varying the relative speed of each rotor to change the thrust produced. It would

be almost impossible to fly these multi-rotor aerial vehicles without an electronic flight controller to stabilize the flight. Even my inexpensive CB180Z has a three-axis gyro system built into the receiver to assist in level flight and turns.

The ELEV-8 uses a HoverFly Sport flight controller board shown in **Figure 7**, with a Propeller multi-core microprocessor to electronically control the stabilization of the aircraft.

Figure 8 shows a modified ELEV-8 with a FPV

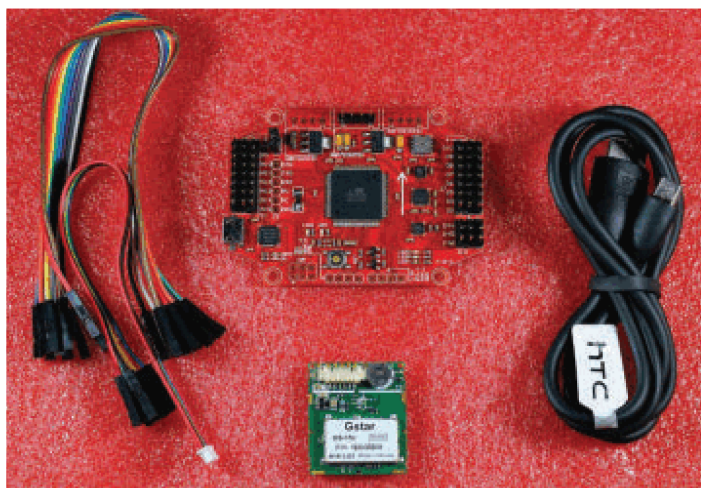


FIGURE 4. MultiWii Pro with MTK GPS module.

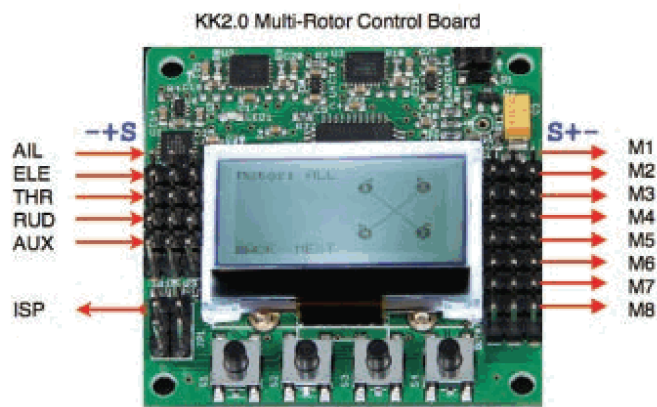


FIGURE 5. The popular KK 2.0 multi-rotor controller.



FIGURE 6. Parallax ELEV-8.

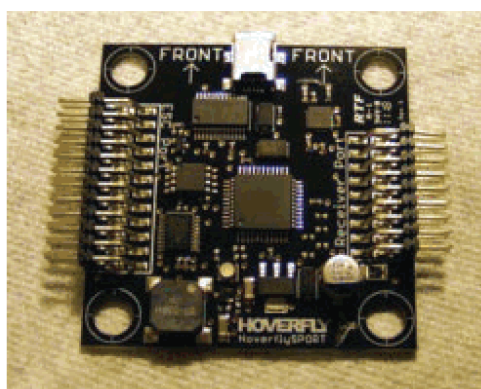


FIGURE 7. HoverFly Sport.

(First Person View) 2.4 GHz directive antenna ground station; there's a cable running to an LCD video display which is out of the picture.

Figure 9 shows two flyers: the test quad mockup on the right with the black whip antenna for the video link, and a hex-modified ELEV-8 in the air. One of Parallax's new versions will utilize a Y-6 configuration with GPS and two-way XBee. "We were inspired to build the

ELEV-8 because the next generation of robotics is clearly taking to the skies, to the water, and in the atmosphere, and we want to be sure that our Propeller multi-core processors are prominently part of the new era of robotics. And there's hardly a better application for a multi-core than Hoverfly's quadcopter controllers. Each process (R/C receiver, ESCs, gyroscope) can be conveniently managed by one of the Propeller's eight cogs," Ken commented.

Ken and his crew have been working on a "really large version of the ELEV-8," among other flying robot projects. Some specifications include two 4 Ahr 18 volt batteries driving four 770 KV motors with 14" props and 80 amp ESCs (electronic

speed controllers). According to Ken, "it will carry lunch for five or a dozen Boe-Bots."

You can get up-to-date information on the ELEV-8 and other multi-rotor developments at the Parallax site or Ken's blog. The original ELEV-8 used four 1,000 KV outrunner brushless motors, turning two pusher and two standard 10 inch props. The large version uses 14 inch props.

The reason for using pushers and standard props in multi-rotor craft is because you don't want the torque from the props to turn the craft. Both types are puller props in that the thrust is downward. Two are turning the opposite direction from the other two to compensate for the rotational torque.

For those not familiar with the KV designation for these types of motors, it does not refer to Kilo-Volts, but to the torque and current draw for a specific voltage and rpm of the motor. In short, the lower the KV rating, the higher the torque.

Electric motors for *winged* model planes have different power designations such as 280, 300, 400, 480, and 600 which refer to the case length, and give an approximation of their power and weight. For example, a 300 motor has about a 30 mm case length, is heavier, and is able to deliver more power than a 280 size motor.

There is a lot of information on the Internet about brushless motors and the unique outrunner motors used



FIGURE 8. Hexcopter and video ground station.



FIGURE 9. Hex and quadcopters flying near Parallax offices.

on most multi-rotor flyers.

Motor Controllers and Electronic Speed Controllers

I've talked with many people about what pieces of equipment they've used and what they like in robot control systems. This includes those who fly the quad/hex rotor-craft, combat robots of all sizes, and medium to large robots that use larger gearmotors to drive the wheels and any appendages. Motors for these types of remotely controlled robots can vary from small 10 watt motors to motors approaching 20 HP or more for large combat robot weapons.

I would like to discuss a few of the transmitter/receiver configurations and the motor controllers used with these remote control systems. Today's choice of system is almost unlimited, with dozens of manufacturers each making dozens of systems and controllers for every type of motor and application that you can imagine.

Vantec Electronic Speed Controllers

I have used Vantec products for over 30 years, and have never had a failure or a performance that did not live up to the specifications. I've used them in promotional robots, service robots, and in action props for movies such as *Revenge of the Nerds* back in early 1984.

The prop people needed four robots for the filming, and they are

not known for their delicate handling of props on a busy set. Even though one was supposedly demolished by the 'jocks' in the movie, the control systems of all four still worked perfectly at the end.

An early RET-4 reversing electronic throttle from Vantec that I used in the early '80s is shown at the lower right of **Figure 10**. Dail DeVilleneuve, founder of the company, has continually expanded and improved his line of R/C products. **Figure 11** shows Dail standing next to one of the *Revenge of the Nerds* robot prop shells in my garage with a 17-channel KeyKoder system in his hands.

The KeyKoder transmitter and receptor system shown in **Figure 10** has an extra receptor board. The KeyKoder allowed the prop people to operate many special effects on the robot by pressing one of 12 buttons on a keypad to operate 12 channels of on-off functions, in addition to five proportional channels. This was an add-on to a standard six-channel R/C system, using one of the "six" for the extra 12 on-off channels. These days — along with supplying combat robot control systems for many winning teams across the country — Vantec also supplies systems for police and military robots, as well as many scientific platforms. The word that I hear back is: "They just keep on working."

I recently tested Vantec's



FIGURE 10. Vantec RET-4 and KeyKoder transmitter and receptor.

RDFR23 dual forward and reverse speed control that is shown in **Figure 12** on a Parallax Stingray robot base (seen in **Figure 13**). This dual ESC allows an R/C operator to use a single stick on their transmitter to control all the movements of a differentially driven (tank style) robot.

Moving the two-channel stick forward or back makes the robot go straight forward or backward. Movement of the stick right or left allows the robot to make shallow turns to complete spinning on the axis. I was drawing close to four amps at 7.2 volts from a LiPo pack which is considerably less than the 30 amp



FIGURE 11. Dail DeVilleneuve and KeyKoder for *Revenge of the Nerds* robot.

capacity of the RDR23, so I clamped two wheelchair motors to a board and drove them at 24 volts at almost 16 amps. Another board lying on the spinning wheels created the drag and load. The ESC got warm, but just kept going. If the RDR23 can run this load for 20 minutes or more, a short combat robot bout at the full peak of 30 amps should be no problem.

As Dail told me, "Heat is the enemy. I use discrete MOSFETs with a larger heatsink base, rather than H-bridges that get very hot."

For longer operational periods, you can screw the case to your robot's metal frame for extra cooling; a fan is not required. I highly recommend Vantec products for their reliability.

Parallax HB-25 Motor Controller

There are many ESCs on the market available for everything from small helicopters to heavy-duty models that can control hundreds of amps. I

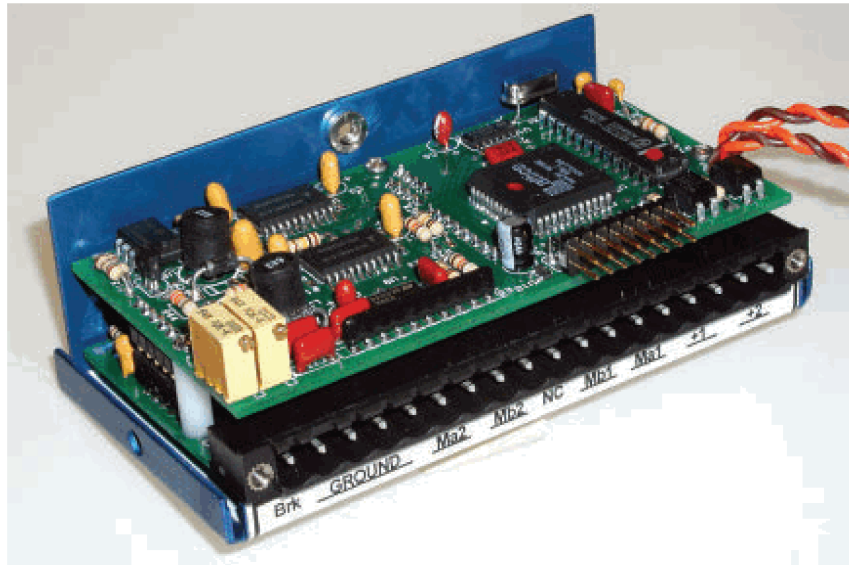


FIGURE 12. Vantec RDR23 dual forward-reverse ESC with lid removed.

won't go into all the other varieties, but I would like to mention the tiny Parallax HB-25 motor controller shown in **Figure 14** that I've come to really like. It's not as tiny as some that are used on quadcopters, but it is quite small for a fan-cooled ESC. **Figure 12** shows one of the two HB-25s on top of the Stingray test base. The blue Vantec RDR shown was used in another test.

With the ability of handling over 400 watts, 16 volts at 25 amps continuous to 35 amps during a surge, this fan-cooled 1-3/4 inch cube ESC is ideal for many mid to large robots.

There's a clearly marked simple three-wire servo input, with two polarized wires for the power input and two for the motor output.

Robot Radio Control Systems

There are three models of radio control systems that seem to be the most popular that I'd like to cover.

Today's systems almost all use the 2.4 GHz band.

Hitec Eclipse 7 Pro System

I have seen Hitec R/C systems used in many combat robots, as well as in large mobile robots. The Eclipse 7 Pro with the accompanying Optima 7 receiver worked for me just as smoothly as Hitec's large line of servos that I have reviewed in the past.

As with most newer systems, these receivers are very tiny as compared with receivers from 20 years ago. **Figure 15**

shows the Eclipse 7 and the accompanying Optima receiver on the left. This is one nice transmitter, both in operation and operating 'feel.' It took me a bit of setup time as I'm not a model airplane driver, and the instruction manual for this and the other R/C radio systems are geared towards flyers.

Setting up the link between the transmitter and receiver — sometimes called 'binding' — was fairly simple, though much more involved than the equipment I used decades ago. Frequency hopping and ID codes are new terms that I am becoming familiar with, but range checks have been necessary since the beginning of R/C.

I tested the system with the receiver connected to four ESCs that were, in turn, connected to four motors spinning freely — not exactly a flying quadcopter. A friend down the street who flies model planes coached me a bit, but it took a day examining the three manuals and experimenting with programming before I felt at ease with these new R/C systems.

The Eclipse 7 Pro feels great in my hands, and operation became intuitive. There are some amazing features too numerous to list here that you should review on the Hitec site. Hitec products have always performed admirably in my projects, and this system proved to be just as reliable

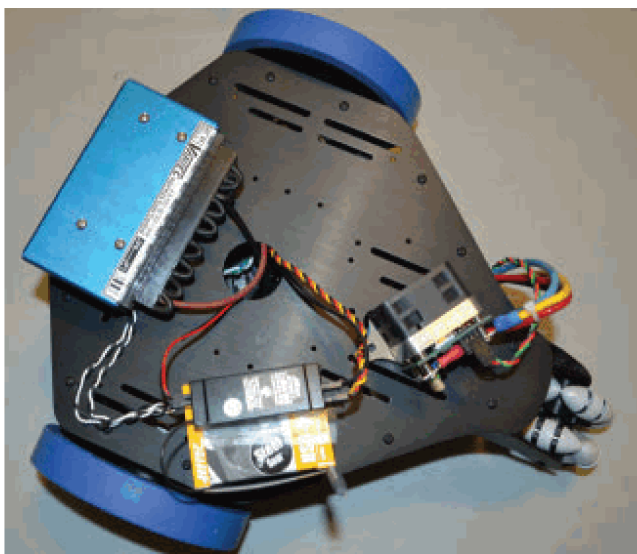


FIGURE 13. Parallax Stingray test base with Vantec RDR23.

and easy to operate.

Spektrum DX7s System

The Spektrum DX7s seven-channel system and its accompanying AR8000 eight-channel receiver shown in the top center of **Figure 15** is a bit more expensive than the Hitec system, but was easy to bind and program, and to connect to my four-motor test setup. I really liked the handling and feel of this system. I did like the placement of the LCD display on the top of the Hitec transmitter better, however.

One feature that startled me at first was the alarm system that both beeped and vibrated the transmitter case when I turned on the TX without verifying the position of the throttle (to zero) and flaps joystick. You don't want your plane or robot leaping away as soon as it receives a signal. In reality, a robot will have two throttle joystick axes, or four or six throttle positions for a quad/hex copter, and no flaps.

If the Hitec system was mid-range in price, then the Spektrum would be high to mid-range, and this was evident in the construction. Most of the multi-rotor flyers that I've seen use Spektrum R/C systems.

When you have an expensive multi-rotor flyer in the air or an expensive combat robot in the arena, you really want a very reliable R/C system linking you to your bot. System information is clearly presented on the screens of these systems — both output signals and status information are relayed back to the transmitter panel. The DX7s is one good system applicable to both ground mobile robots, as well as multi-rotor craft.

Futaba T6J System

I would class the Futaba T6J shown at the top right of **Figure 15** as a low mid-range R/C system in price, however, it performed quite well during testing. It did not come with batteries or a charger like the other two systems, but it had every feature that any robot or multi-rotor craft builder would want.

I immediately felt at ease with the TJ6 and the accompanying R2006GS receiver, possibly because I have used Futaba equipment in the past. The Vantec KeyKoder system back in **Figure 10** used a Futaba radio and neither of the two systems failed me.

The linking procedure for the system was easy. I didn't test the Futaba T6J on my four-motor setup but rather hooked it to the Parallax Stingray and tormented my cat as the robot darted about using two of the six channels.

I actually didn't like the smooth feel of the sides as I could see myself dropping the transmitter at some point, but a neck lanyard was furnished with this TX, as well as the others. The LCD display was smaller, but still showed all the critical functions.

The antenna was concealed in the top, but a 2.4 GHz antenna does not need to be large. Its range was not as far as the other two, but easily transmitted down my street approximately 250 feet. This system was also designed for model aircraft, but can easily be converted for robot control. I would recommend any of these three systems for a robot builder

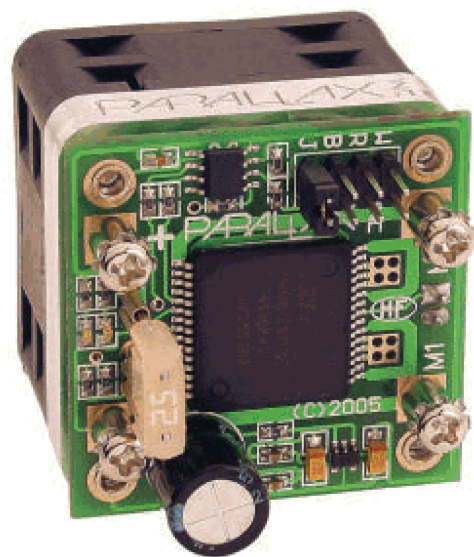


FIGURE 14. Parallax HB-25 fan-cooled electronic speed controller.

or multi-rotor flyer.

Final Thoughts

I have only touched upon a few selected systems and components here, but hope I've piqued your interest. I highly recommend that you read about all the different types of systems and talk with those who have used them. Michael Simpson's article series is a great starting point. Good luck with whatever type of robot project you choose, whether it's one that flies or runs on the ground. **SV**



FIGURE 15. Hitec, Spektrum, and Futaba R/C systems, plus classic systems from the '80s.

ROBO-LINKS

RobotShop
.com



Pololu
Robotics & Electronics
WWW.POLOLU.COM



www.canakit.com

- Over 200 Kits & Modules to choose from!
- Easy Worldwide Online Ordering
- Fast Delivery & Same Day Shipping

FM Transmitters • Timers • Audio Amplifiers • Motor Controllers



NEW!
SERVO POWER GEARBOXES

- Full metal gears
- Up to **3,402 oz-in** of torque
- Fits inside our channel
- 90°, 180°, or 360° rotation available!

www.servocity.com



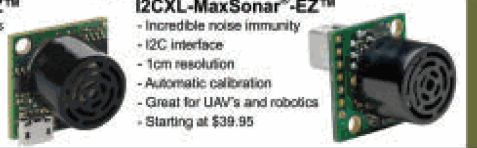
Ultrasonic Distance Sensing Made EZ www.maxbotix.com

USB-ProxSonar®-EZ™

- Great for kiosk applications
- USB interface
- Easy integration
- 1 inch resolution
- Calibrated beam pattern
- Starting at \$49.95

I2CXL-MaxSonar®-EZ™

- Incredible noise immunity
- I2C interface
- 1cm resolution
- Automatic calibration
- Great for UAV's and robotics
- Starting at \$39.95



THE ORIGINAL SINCE 1994
PCB-POOL®
Beta LAYOUT

- Low Cost PCB prototypes
- Free laser SMT stencil with all Proto orders

WWW.PCB-POOL.COM



AndyMark
Inspiring Mobility

www.andymark.com



ALL ELECTRONICS CORPORATION
Electronic Parts & Supplies
Since 1967



INVEST in your BOT!

HITEC

12115 Paine Street • Poway, CA 92064 • 658-748-6948 • www.hiteccrd.com



SERVO MAGAZINE

2006, 2007, 2008, 2009, 2010, 2011, 2012

Free Shipping!

9 CD-ROMs & Hat Special

Only \$ 189.95 or \$24.95 each.



For the finest in robots, parts, and services, go to
www.servomagazine.com and
click on Robo-Links.

ADVERTISER INDEX

All Electronics Corp.21, 80	Maxbotix80	Servo City/Robot Zone80, 83
AndyMark63, 80	PanaVise63	Solutions Cubed7
AP Circuits73	PCB Pool63, 80	Trenton Computer Festival52
Cana Kit Corp. 80, 82	Pololu Robotics & Electronics ...3, 80	Univ. for Advancing Technology ..35
DigilentBack Cover	RoboGames34	Vantec63
Fast Protos.com15	Robotis53	Weirdstuff Warehouse21
Firgelli14	RobotShop, Inc43, 80, 81	
HiTec2, 80	SDP/SI21	

ROBOTSHOP WILL INTRODUCE...

— FLOWBOTICS STUDIO V2 —

05.24.2013

GET READY TO CHANGE THE WORLD OF ROBOTICS



THE COUNTDOWN HAS BEGUN

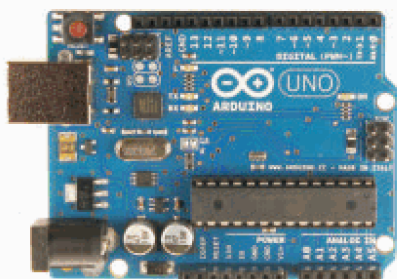
FlowBotics is a trademark of
RobotShop Distribution inc.

RobotShop.com
Robotics at Your Service!™



Motor Controllers • Arduino • Arduino Kits • Arduino Shields • Robot Platforms • Parts

Arduino Uno



\$29.95

ARDUINO-UNO
Latest "R3" Version!

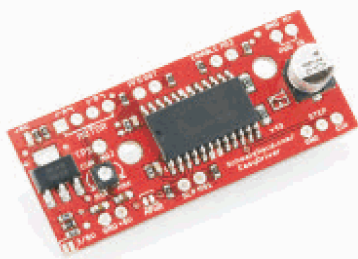
Inventor's Kit for Arduino
ARDUINO-PRK



\$94.95

Includes 36-Page Printed Full-Color Manual

EasyDriver Stepper Motor Driver



SX09402

\$14.95

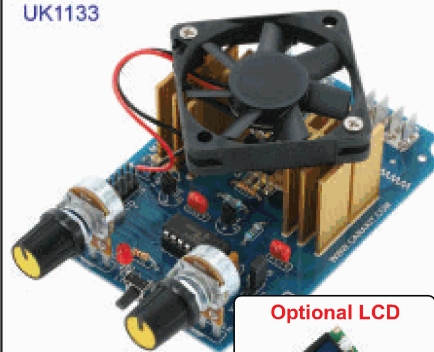
Stepper Motor



SX09238

\$19.95

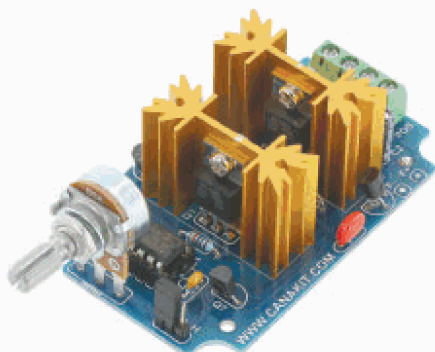
50A Motor Speed Controller (PWM)
UK1133



Optional LCD

\$59.95

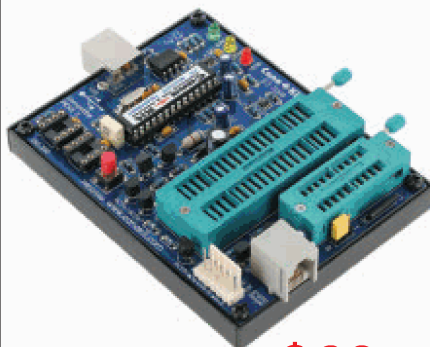
7A Bi-Directional Motor Speed Controller (PWM)



UK1125

\$39.95

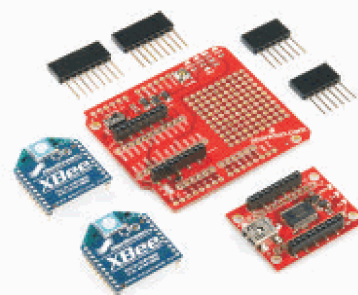
USB PIC Programmer (MPLAB Compatible)



UK1300

\$69.95

Xbee Wireless Kit



XBEE-KIT

\$94.95

Introducing our **NEW** robotic components!



Servo Power Gearboxes!

- Full metal gears
- Up to **3,402 oz-in** of torque
- Fits inside aluminum channel
- Digital and programmable

90°, 180°, 360°
or continuous
rotation options
available!



#615138 #585444 #535114

#615274 #575148 #615290

#545348 #633104-16 #585404

#585426 #575114 #615222

#535130 #545456

#585454 #545448 #595614

#525124 #525134 #535142

#585436 #545324 #585496

#545340 #615126 #535182

#625102 #585403 #545440

#545408 #545392 #545412

#545432 #535162 #545388

#545344 #535134 #545354

#585504 #585500 #585506

#585484 #555132 #555124

#545488 #585482 #585470

#595612 #545424 #545492

#585472 #545452 #585434

#585468 #615254 #585484

#615114 #545420 #585428

#545360 #585494 #585474

#535134 #545354 #585422

#545352 #545496 #535112

#615198 #625106 #615238

#615242 #615130 #615190

#635166 #545404 #585462

#585476 #555128 #545356

#545444 #545436 #545396

#585430 #585443 #535150

#575124 #525130 #555116

#545368 #585424 #535170

#545364 #545416 #575112

#535138 #585478 #545468

#535110 #585488

www.ServoCity.com

We have the parts for your ideas - shop online today!

Find us on:

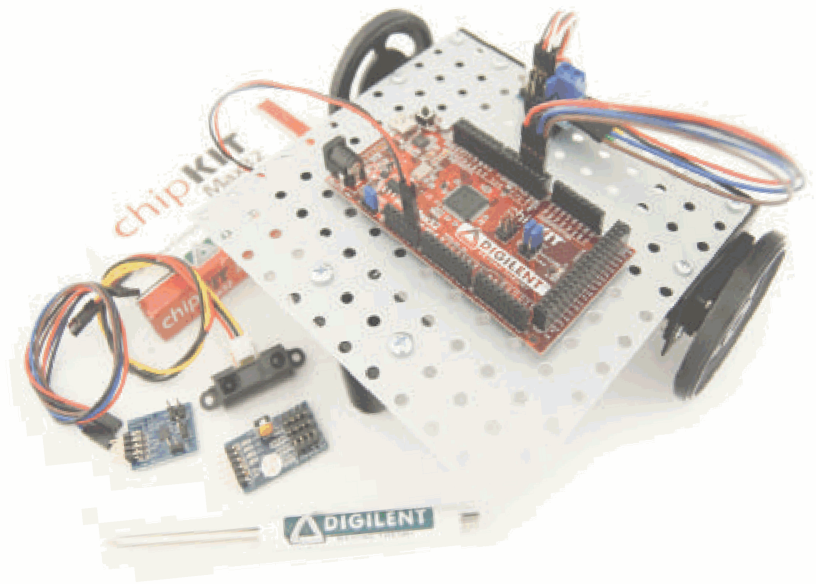


...and so much more!

View our entire
selection of quality
products online



ADVANCED 32-BIT POWER FOR THE ARDUINO COMMUNITY



chipKIT™ development boards are the first 32-bit microcontroller-based platforms that are compatible with many existing Arduino™ code examples, reference materials and other resources. They can be programmed using an environment based on the original Arduino™ IDE modified to support PIC32.

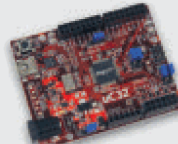
- Pin-out compatibility with many existing Arduino™ shields that can operate at 3.3V
- Lower price-point at four times the performance than existing solutions
- Advanced capabilities including integrated USB (Device/Host, OTG) & integrated Ethernet

chipKIT™ Max32 \$49.50



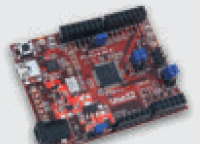
- Microchip® PIC32MX795F512 processor
80 Mhz 32-bit MIPS
512K Flash, 128K RAM
USB 2.0 OTG controller
10/100 Ethernet MAC
Dual CAN controllers
- Can also be programmed in Microchip's MPLAB
- Arduino™ "Mega" form factor
- Compatible with many Arduino™ shields
- 83 available I/O
- User LED

chipKIT™ uC32 \$34.99



- Microchip® PIC32MX340F512H microcontroller
80 Mhz 32-bit MIPS
512K Flash, 32K SRAM
- Can also be programmed in Microchip's MPLAB
- Arduino™ "Uno" form factor
- Compatible with many Arduino™ shields
- 42 available I/O
- Two user LEDs

chipKIT™ Uno32 \$26.95



- Microchip® PIC32MX320F128 processor
80 Mhz 32-bit MIPS
128K Flash, 16K SRAM
- Can also be programmed in Microchip's MPLAB
- Arduino™ "Uno" form factor
- Compatible with many Arduino™ shields
- 42 available I/O
- User LED

Network™ Shield \$54.99



- Usable with the Max32
- 10/100 Ethernet
- USB Host, Device, OTG
- Dual CAN transceivers
- Dual I2C™ connectors
- 256kbit I2C™ EEPROM
- 32.768 Khz oscillator for RTCC

Basic I/O™ Shield \$37.99



- Usable with the Max32, Uno32, & uC32
- 128x32 OLED Graphic Display
- Digital temperature sensor
- 256kbit EEPROM
- 4 switches, 4 push buttons, 8 LEDs
- 4 Open drain transistor outputs
- Analog potentiometer

Pmod 8™ Shield \$21.99



- Usable with the Uno32
- Five 2x6-pin Digilent Pmod connectors
- One 6-pin SPI connector
- One I²C daisy chain connector

WiFi™ Shield \$49.99



- Usable with the Max32, Uno32, & uC32
- IEEE 802.11b-compliant RF transceiver
- 1 and 2Mbps data rates
- IEEE 802.11b/g/n-compatible
- Integrated PCB antenna
- Micro SD card connector
- Four LEDs

chipKIT™ PGM \$26.99



The chipKIT PGM is a simple, low cost, module that supports in-system programming and debugging of applications written for Microchip PIC based microcontroller boards such as the chipKIT™ and Cerebot boards. It is designed to work with the MPLAB® and MPLAB® X development environments available from Microchip.

The chipKIT PGM can also be used to enable in-system debugging of sketches developed using the Arduino™ compatible MPIDE development environment.

chipKIT™



www.digilentinc.com/chipkit